

August 1978

This document provides the information needed to use the VAX/VMS MCR command language, execute MCR indirect command files, and use RSX-11M Version 3.1 components under VAX/VMS.

VAX-11/RSX-11M

User's Guide

Order No. AA-D037A-TE

SUPERSESSON/UPDATE INFORMATION: This is a new document for this release.

OPERATING SYSTEM AND VERSION: VAX/VMS V01

SOFTWARE VERSION: VAX/VMS V01

To order additional copies of this document, contact the Software Distribution Center, Digital Equipment Corporation, Maynard, Massachusetts 01754

digital equipment corporation • maynard. massachusetts

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

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

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

Copyright © 1978 by Digital Equipment Corporation

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

The following are trademarks of Digital Equipment Corporation:

DIGITAL	DECsystem-10	MASSBUS
DEC	DECTape	OMNIBUS
PDP	DIBOL	OS/8
DECUS	EDUSYSTEM	PHA
UNIBUS	FLIP CHIP	RSTS
COMPUTER LABS	FOCAL	RSX
COMTEX	INDAC	TYPESET-8
DDT	LAB-8	TYPESET-11
DECCOMM	DECSYSTEM-20	TMS-11
ASSIST-11	RTS-8	ITPS-10
VAX	VMS	SBI
DECnet	IAS	

CONTENTS

	Page
PREFACE	vii
CHAPTER 1 INTRODUCTION	1-1
1.1 WHAT IMAGES RUN?	1-1
1.1.1 Requirements for RSX-11M Image Execution	1-2
1.2 WHAT RSX-11M PROGRAM DEVELOPMENT TOOLS ARE AVAILABLE?	1-2
1.2.1 Standard Utilities and Files Provided with VAX/VMS	1-2
1.2.2 Additional Utilities and Programs	1-3
1.3 FILES-11	1-3
1.4 FCS AND RMS-11 UNDER VAX/VMS	1-4
1.5 SUPPORT OF RSX-11M DIRECTIVES	1-4
1.6 SUMMARY OF COMMANDS	1-4
CHAPTER 2 VAX/VMS ENVIRONMENT	2-1
2.1 PROCESS AND IMAGE	2-1
2.1.1 Process and Image versus Task	2-1
2.1.2 Login Process, Detached Process, or Subprocess	2-2
2.1.3 Process Name	2-2
2.1.4 Process Identification	2-2
2.2 USER AUTHORIZATION FILE	2-3
2.2.1 VAX/VMS UICs and Protection	2-3
2.2.2 Privileges	2-4
2.2.3 Process Priority	2-5
2.3 FILE SPECIFICATIONS	2-5
2.3.1 Device Names	2-6
2.3.2 Directory Names	2-7
2.3.3 Version Numbers	2-8
2.3.4 Defaults in File Specification	2-8
2.4 LOGICAL NAMES	2-10
2.4.1 Process Logical Name Table	2-10
2.4.2 Group Logical Name Table	2-10
2.4.3 System Logical Name Table	2-11
2.4.4 System-Defined Logical Names	2-11
2.5 DEVICE MAPPING	2-12
2.5.1 Mapping RSX-11M Pseudo-Device Names	2-12
2.5.2 Mapping RSX-11M Physical Device Names	2-13
2.6 TERMINAL FUNCTIONS	2-13
2.7 SYNTAX RULES	2-15
2.7.1 Continuing Commands on More than One Line	2-15
2.7.2 Commenting Command Lines	2-16
2.7.3 Truncating Command Names and Keywords	2-16
2.7.4 Placement of Keywords	2-16
2.7.5 Specifying File Protection	2-17
2.7.6 Entering Dates and Times	2-18
2.8 ERROR MESSAGES	2-20

CONTENTS (Cont.)

	Page
CHAPTER 3 USING VAX/VMS MCR	3-1
3.1 LOGGING INTO VAX/VMS	3-1
3.1.1 Selecting a Command Interpreter	3-2
3.1.2 LOGIN.CMD File	3-2
3.2 PREPARING TO RUN AN IMAGE	3-2
3.2.1 Allocating Devices	3-2
3.2.2 Mounting Volumes	3-3
3.2.3 Assigning Logical Names	3-4
3.2.4 Installing RSX-11M Images	3-5
3.3 RUNNING IMAGES	3-6
3.4 PROGRAM DEVELOPMENT UNDER VAX/VMS	3-6
3.4.1 Using RSX-11M Components	3-7
3.4.2 Using FLX	3-9
3.4.3 Using VAX/VMS Components	3-10
3.5 EXECUTING RSX-11M INDIRECT COMMAND FILES	3-10
3.6 BUILDING RSX-11M IMAGES FOR EXECUTION UNDER VAX/VMS	3-10
3.6.1 Task Builder Switches	3-11
3.6.2 Task Builder Options	3-12
3.6.3 Noncontiguous Image Files	3-12
3.6.4 Building RSX-11M Images on VAX/VMS for Execution on RSX-11M	3-12
CHAPTER 4 MCR COMMANDS	4-1
4.1 ALLOCATE DEVICE	4-1
4.2 APPEND	4-3
4.3 ASSIGN	4-6
4.4 BYE	4-8
4.5 CANCEL	4-9
4.6 CONTINUE	4-11
4.7 COPY	4-12
4.8 DEALLOCATE	4-16
4.9 DEBUG	4-17
4.10 DELETE	4-18
4.11 DEPOSIT	4-21
4.12 DIRECTORY	4-24
4.13 DISMOUNT	4-27
4.14 EXAMINE	4-29
4.15 EXECUTE PROCEDURE (@)	4-32
4.16 EXIT	4-34
4.17 INITIALIZE	4-36
4.18 LINK	4-43
4.19 LOGIN PROCEDURE	4-49
4.20 LOGOUT	4-50
4.21 MOUNT	4-51
4.22 ON STATEMENT	4-59
4.22.1 On Control Y	4-60
4.22.2 On Severity-level	4-62
4.23 PRINT	4-64
4.24 PURGE	4-68
4.25 RUN	4-69
4.26 SET	4-79
4.26.1 Set Card Reader	4-81
4.26.2 Set Control Y	4-82
4.26.3 Set Default	4-83
4.26.4 Set Magtape	4-84

CONTENTS (Cont.)

	Page	
4.26.5	Set On	4-85
4.26.6	Set Process	4-86
4.26.7	Set Protection	4-87
4.26.8	Set Queue	4-88
4.26.9	Set RMS Default	4-89
4.26.10	Set Terminal	4-90
4.26.11	Set Default UIC	4-92
4.26.12	Set Verify	4-93
4.26.13	Set Working Set	4-94
4.27	SHOW	4-95
4.27.1	Show Daytime	4-96
4.27.2	Show Default	4-97
4.27.3	Show Devices	4-98
4.27.4	Show Logical	4-99
4.27.5	Show Magtape	4-100
4.27.6	Show Network	4-101
4.27.7	Show Printer	4-102
4.27.8	Show Process	4-103
4.27.9	Show Protection	4-104
4.27.10	Show Queue	4-105
4.27.11	Show RMS Default	4-106
4.27.12	Show Status	4-107
4.27.13	Show Symbols	4-108
4.27.14	Show System	4-109
4.27.15	Show Terminal	4-110
4.27.16	Show Translation	4-111
4.27.17	Show Working Set	4-112
4.28	STOP	4-113
4.29	SUBMIT	4-115
4.30	TIME	4-118
4.31	TYPE	4-119
4.32	UFD	4-120
CHAPTER 5	INDIRECT COMMAND FILES	5-1
5.1	REQUESTING INDIRECT COMMAND FILE EXECUTION	5-6
5.2	SUPPORT OF .IFINS, .IFACT, AND .IFLOA	5-6
5.3	SUPPORT OF .ENABLE AND .DISABLE ESCAPE	5-6
5.4	UNSUPPORTED DIRECTIVES	5-6
5.4.1	.XQT and .WAIT Alternative	5-7
5.4.2	.ONERR Alternative	5-7
5.5	SWITCHES	5-7
5.6	SPECIAL SYMBOLS	5-7
5.7	NUMERIC SYMBOLS	5-8
5.8	LABEL TABLE ENTRIES	5-9
5.9	SYS\$INPUT AND SYS\$COMMAND	5-9
5.10	.ASKN DIRECTIVE	5-10
5.11	.ENABLE DOLLARS DIRECTIVE	5-10
5.12	.PAUSE DIRECTIVE	5-11
5.13	.DELAY DIRECTIVE	5-11
5.14	NESTING MCR INDIRECT COMMAND FILES	5-11
5.15	LEXICAL FUNCTIONS	5-12
5.16	OPERATIONAL NOTES	5-13
5.17	RSX-11M AND RSX-11S SYSTEM GENERATION	5-14
5.17.1	Creation of an Unprotected Disk	5-14
5.17.2	Pre-System Generation Indirect Command File	5-14

CONTENTS (Cont.)

		Page
APPENDIX A	PRIVILEGES	A-1
APPENDIX B	RESOURCE QUOTAS AND LIMITS	B-1
FIGURES		
FIGURE	2-1	2-3
	3-1	3-7
	5-1	5-15
TABLES		
TABLE	1-1	1-4
	2-1	2-7
	2-2	2-8
	2-3	2-9
	2-4	2-12
	2-5	2-14
	3-1	3-11
	4-1	4-79
	4-2	4-91
	4-3	4-95
	5-1	5-2
	5-2	5-8
	5-3	5-12

PREFACE

MANUAL OBJECTIVES

The VAX-11/RSX-11M User's Guide provides the information needed to use the VAX/VMS MCR command language, execute MCR indirect command files, and use RSX-11M Version 3.1 components (for example, the task builder) under VAX/VMS. The information in this document is intended to allow RSX-11M users to operate in a familiar environment, while at the same time encourage users to make the transition to the DIGITAL Command Language (DCL), the primary VAX/VMS command language.

INTENDED AUDIENCE

This manual contains the information that an RSX-11M user needs to run existing RSX-11M Version 3.1 images and native VAX-11 images under VAX/VMS, and to develop both RSX-11M and native VAX-11 images using the VAX/VMS MCR command language.

This document has two prerequisites: a general understanding of RSX-11M and an understanding of the material presented in either the VAX/VMS Summary Description or the VAX-11/780 Technical Summary.

STRUCTURE OF THIS DOCUMENT

Information in this document is organized as follows.

- Chapter 1 describes the requirements for executing RSX-11M images under VAX/VMS and lists the RSX-11M components available under VAX/VMS.
- Chapter 2 discusses the environment provided by VAX/VMS, and describes command language features and syntax rules.
- Chapter 3 describes how to log into the system, select a command interpreter, prepare images for execution, run user images (both RSX-11M and VAX-11), and use RSX-11M and VAX/VMS components for RSX-11M and VAX-11 program development.
- Chapter 4 describes the VAX/VMS MCR commands in detail.
- Chapter 5 describes the VAX/VMS MCR directives and provides information on performing an RSX-11M/S system generation using VAX/VMS as a host system.
- Appendix A describes user privileges, and Appendix B describes resource usage limits and quotas.

ASSOCIATED DOCUMENTS

The following additional documents may also be useful.

- VAX-11 Information Directory
- VAX-11/RSX-11M Programmer's Reference Manual
- VAX/VMS Primer
- VAX/VMS Command Language User's Guide

Of the documents listed above, the VAX/VMS Command Language User's Guide provides the most useful additional information. The VAX/VMS MCR command language has many features in common with the primary VAX/VMS command language, DCL. The VAX-11/RSX-11 User's Guide describes these features to the extent needed to perform the VAX/VMS equivalent of normal RSX-11M functions. You can learn about additional capabilities available with MCR by reading Part I of the VAX/VMS Command Language User's Guide.

CONVENTIONS USED IN THIS DOCUMENT

In the command and directive descriptions in Chapters 4 and 5, respectively, square brackets ([]) indicate optional syntax. Brackets that are part of directory names, however, do not indicate optional syntax. In addition, certain MCR directives use brackets as part of their required syntax; these cases are noted in the text.

CHAPTER 1

INTRODUCTION

VAX/VMS provides two command languages you can use to interface with the system:

- DIGITAL command language (DCL), as described in the VAX/VMS Primer and the VAX/VMS Command Language User's Guide
- MCR command language, as described in this document

The VAX/VMS MCR command language is an integral part of the VAX/VMS operating system. It allows RSX-11M users to make the transition to VAX/VMS using a familiar command language. VAX/VMS MCR and indirect command file capability also permit use of the VAX/VMS system as the host for RSX-11M/S Version 3.1 system generation.

The VAX/VMS MCR command language consists of two types of commands:

- Those that duplicate an RSX-11M command
- Those that provide a VAX/VMS function using an MCR-like syntax

Because of the two types of commands, MCR allows access to a full range of VAX/VMS functions. You need not change to the DCL command language to perform commonly needed functions.

1.1 WHAT IMAGES RUN?

VAX/VMS MCR provides the tools needed to develop and run both RSX-11M Version 3.1 task images and native VAX-11 images. This capability means that, as user-written applications gradually convert to native mode, applications consisting of both RSX-11M and native images can be handled using one command language. Both types of images can also be run from one indirect command file using MCR.

VAX/VMS places no restrictions on the native images that you can run using MCR; the system manager imposes any restrictions that exist on a user-by-user basis. The VAX-11/780 processor and the VAX/VMS operating system impose restrictions on the RSX-11M images that can execute. The restrictions at both the hardware and software levels are described in the VAX-11/RSX-11M Programmer's Reference Manual and summarized below.

INTRODUCTION

1.1.1 Requirements for RSX-11M Image Execution

At the processor level, the requirements for RSX-11M image execution are as follows.

- The image must execute in user mode.
- It must not execute any privileged instructions such as HALT or RESET.
- It must not issue FIS floating point instructions; the software emulates FPP floating point instructions.

At the software level, the requirements are as follows.

- The image must have been built using the RSX-11M Version 3.1 task builder to execute in a mapped system.
- It must not depend on environmental features of RSX-11M that are not available in VAX/VMS, for example, PLAS memory management or significant events.
- It must not overlap the RSX-11M executive; the RSX-11M executive is not present in VAX/VMS.
- It must not overlap the PDP-11 I/O page; the I/O page is not available at user mode.

Task images developed under RSX-11D or IAS and compatible with RSX-11M can execute under VAX/VMS if they meet the requirements listed above. However, such task images must be rebuilt using the RSX-11M Version 3.1 task builder before they can execute under VAX/VMS.

RSX-11M Version 3.1 task images do not have to be rebuilt to run under VAX/VMS unless program modification or different task builder options are required. Rebuilding also is required to take advantage of the logical name extensions of FCS and RMS-11.

1.2 WHAT RSX-11M PROGRAM DEVELOPMENT TOOLS ARE AVAILABLE?

VAX/VMS provides the RSX-11M utilities needed for the operating system to function as the host for RSX-11M program development. These utilities run in compatibility mode under VAX/VMS and provide functions that are identical to the functions they provide in RSX-11M. In addition, VAX/VMS provides copies of the files and libraries needed for program development.

1.2.1 Standard Utilities and Files Provided with VAX/VMS

VAX/VMS provides the following RSX-11M utilities as part of the standard system.

- MACRO-11 (MAC)
- Editor (EDI)
- RSX-11M task builder (TKB)
- RSX-11M librarian (LBR)

INTRODUCTION

- Peripheral interchange program (PIP)
- File transfer utility (FLX)
- Source language input program (SLP)
- File structure verification utility for Files-11 Structure Level 1 (VFY1)
- ZAP utility (ZAP)
- File patch utility (PAT)
- Disk save and compress utility for Files-11 Structure Level 1 (DSC1)
- File dump utility (DMP)
- Bad block locator utility (BAD)
- RMS-11 Utilities

VAX/VMS also provides the following RSX-11M files and libraries.

- SYSLIB.OLB
- RSXMAC.MLB
- ODT.OBJ

With the utilities and files listed above, you can use VAX/VMS as you would RSX-11M for program development.

For example, you can enter and edit a program using EDI, assemble it using MACRO-11, build it using the RSX-11M task builder, and run it using VAX/VMS as the host system. Images that are to be debugged using ODT can be linked with ODT.OBJ.

RSX-11M task images developed under VAX/VMS can execute under RSX-11M.

1.2.2 Additional Utilities and Programs

The following additional RSX-11M facilities are available with VAX/VMS under separate license:

- RSX-11M/S system generation
- FORTRAN IV compiler

1.3 FILES-11

Files-11 is the file structure used for disk volumes under both RSX-11M and VAX/VMS. Files-11 provides two structure levels: Files-11 Structure Level 1 and Files-11 Structure Level 2. RSX-11M supports only structure level 1, whereas VAX/VMS supports both structure levels. By default, VAX/VMS creates structure level 2 volumes; however, it can create and read structure level 1 volumes. Thus, structure level 1 volumes can be transported among RSX-11, IAS, and VAX/VMS systems.

INTRODUCTION

1.4 FCS AND RMS-11 UNDER VAX/VMS

File Control Services (FCS), RMS-11, and RMS-11K (ISAM) are available under VAX/VMS, and all run in compatibility mode. The result is that RSX-11M images that call FCS/RMS-11 interface with the same record management services that they used in RSX-11M. VAX/VMS converts FCS and RMS-11 requests from the RSX-11M format to their VAX/VMS equivalent format. The VAX-11V RSX-11M Programmer's Reference Manual details the conversion process.

Record locking, however, is not supported.

1.5 SUPPORT OF RSX-11M DIRECTIVES

The VAX-11/RSX-11M Programmer's Reference Manual describes VAX/VMS support for RSX-11M task images that issue directives to the RSX-11M executive. The majority of directives are supported by VAX/VMS; the following, however, are not:

- PLAS directives because of substantial differences in hardware memory management techniques
- GET SENSE SWITCH because VAX-11/780 does not have sense switches
- CONNECT TO INTERRUPT VECTOR because I/O drivers do not run in compatibility mode under VAX/VMS.

1.6 SUMMARY OF COMMANDS

Table 1-1 summarizes the VAX/VMS MCR commands. In addition to the command listed, VAX/VMS MCR recognizes commands that invoke RSX-11M components, for example, MAC to invoke the MACRO-11 assembler.

Table 1-1
Summary of MCR Commands

Command	Function
Initialization Commands	
Allocate	Reserves an unmounted shareable device or an unallocated nonshareable device for exclusive use by the process
Assign	Defines or deletes a logical name assignment
Deallocate	Releases a previously allocated device
Dismount	Releases a volume previously specified in a Mount command
Initialize	Initializes a Files-11 Structure Level 1 or 2 disk volume

(continued on next page)

INTRODUCTION

Table 1-1 (Cont.)
Summary of MCR Commands

Command	Function
Mount	Makes a disk or magnetic tape volume or volume set available for processing
Set Card Reader	Defines the default ASCII translation mode for a card reader
Set Control Y	Enables recognition of CTRL/Y to return to the command interpreter
Set Default	Establishes a disk and/or directory as the current default for file specifications
Set Default UIC	Establishes a new default directory and UIC
Set Magtape	Defines the characteristics of a magnetic tape device
Set On	Requests the command interpreter to check for an error condition after executing a command or program in an indirect command file
Set Process	Defines execution characteristics of the current process
Set Protection	Defines the protection status of a file or group of files, or establishes the default protection for files created subsequently during the job
Set Queue	Changes the attributes associated with one or more entries in a printer or batch job queue
Set RMS Default	Provides default multiblock and multibuffer counts to be used by VAX-11 RMS for file operations; RMS-11 does not use these defaults
Set Terminal	Defines operational characteristics of a terminal
Set /UIC	Changes the default directory and the UIC of the current process
Set Verify	Requests the command interpreter to display lines in indirect command files as it executes them
Set Working Set	Changes the current working set limit or quota
UFD	Creates a directory or subdirectory on a Files-11 Structure Level 1 or 2 disk

(continued on next page)

INTRODUCTION

Table 1-1 (Cont.)
Summary of MCR Commands

Command	Function
Informational Commands	
Show Daytime	Displays the current date and time
Show Default	Displays the current default disk and directory
Show Devices	Displays the status of devices in the system
Show Logical	Displays current logical name assignments
Show Magtape	Displays the characteristics of a magnetic tape device
Show Network	Displays the availability of network nodes, including the current node
Show Process	Displays attributes of the current process, including privileges, resource quotas, memory usage, priority, and accounting information
Show Protection	Displays the current default protection applied to files
Show Queue	Displays the status of printer or batch jobs that have been queued but not yet printed or processed
Show RMS Default	Displays the current default multiblock and multibuffer counts that VAX-11 RMS uses for file operations
Show Status	Displays the status of the current job, including accumulated CPU time, open file count, and counts of I/O operations
Show Symbols	Displays current symbol definitions
Show System	Displays the status of all processes in the system
Show Terminal	Displays the device characteristics of your terminal
Show Translation	Displays the equivalence name of a specified logical name
Show Working Set	Displays the current working set size limit and quota
Time	Displays the current date and time

(continued on next page)

INTRODUCTION

Table 1-1 (Cont.)
Summary of MCR Commands

Command	Function
Process Control Commands	
Bye	Terminates an interactive terminal session
Cancel	Cancels scheduled wakeup requests for a process
Continue	Resumes execution of an interrupted image or indirect command file
Exit	Terminates processing of the current indirect command file
On Control Y	Defines the course of action to be taken when CTRL/Y is pressed
On severity-level	Defines the default course of action to be taken when a command or program in an indirect command file encounters an error condition
Login procedure	Verifies your right to use the system and logs you into it
Logout	Terminates an interactive terminal session
Run	Runs images and, optionally, creates detached processes and subprocesses
Stop	Terminates execution of the image currently running in a process
Submit	Queues a batch job for processing
Program Development Commands	
Debug	Invokes a debugger for an interrupted image
Deposit	Replaces the content of specified locations of virtual memory
Examine	Examines the content of specified locations of virtual memory
Link	Links native VAX-11 images
File Manipulation Commands	
Append	Adds the contents of one or more input files to the end of an output file
Copy	Creates new files from one or more existing files

(continued on next page)

INTRODUCTION

Table 1-1 (Cont.)
Summary of MCR Commands

Command	Function
Delete	Deletes files, entries from printer queues, and entries from batch job queues
Directory	Displays the names of one or more files in a directory or subdirectory
Print	Queues files for printing
Purge	Deletes older versions of a file while saving the specified number of recent versions
Type	Displays the contents of a file at your terminal

CHAPTER 2

VAX/VMS ENVIRONMENT

When using VAX/VMS MCR, you will be aware that the environment provided by this operating system differs from that provided by RSX-11M. For example, you have access to features not available in RSX-11M, and device names are different. This chapter describes some basic VAX/VMS concepts, operational facts, and syntax rules of which you should be aware.

2.1 PROCESS AND IMAGE

When you log into VAX/VMS, the system automatically creates a process for you. That process provides the virtual address space and control information needed to perform all normal interactive user operations. User requests are performed by the execution of a series of images. Under VAX/VMS, an image is a program that has been made executable as a result of linking by either the VAX-11 Linker or the RSX-11M Task Builder (TKB). The VAX-11 Linker is an example of a native image; PIP and TKB are examples of RSX-11M images. You can run both native and RSX-11M images within your process using MCR.

Images in VAX/VMS, like tasks in RSX-11M, reside in files on disk. When you request the execution of an image (for example, by typing the RUN command), VAX/VMS activates that image in your process. As each subsequent image is requested, it overlays the previous image in the process's virtual address space. The VAX/VMS Summary Description further describes the concepts of process and image. The sections that follow describe the features of a process that are significant for an RSX-11M user.

2.1.1 Process and Image versus Task

A process executing an image under VAX/VMS is quite similar in concept to an executing RSX-11M task. However, the following differences exist.

- A process is permanently associated with the user for which it was created; it is not bound to a particular image.
- A process can serially execute any number of images.
- A process normally remains until the user logs off the system.

2.1.2 Login Process, Detached Process, or Subprocess

VAX/VMS recognizes three types of processes.

- A login process; that is, the process created for you when you log into the system
- A detached process; that is, a fully independent process
- A subprocess; that is, a process that you create to run a specified image, but which remains under control of your process

Normally, you run images in your login process. Because that process is associated with you and your terminal, you can request only one image at a time. In most program development sessions, where the output from one step (for example, MACRO-11) is input to the next image (for example, the task builder), this serial approach is convenient. However, in an application where images must run concurrently to coordinate their activities, each image must run in a separate process. Therefore, the MCR Run command allows you to create subprocesses and detached processes, in addition to running images in your current process.

2.1.3 Process Name

Every process has a process name that is created as a result of one of the following events:

- When you log in, VAX/VMS defines a process name that is your user name, for example, FELIX.
- A native image running in your process can issue a system service request.
- If you run an RSX-11M image, VAX/VMS creates a process name if the image has a task name that starts with an alphabetic character in its image label block. Utilities do not cause the process name to change because these names start with ..., for example, ...PIP.

Normally, a process name remains for the duration of the process or until a new name is specified. When you run an RSX-11M image that has a task name starting with an alphabetic character, however, the task name becomes the process name for the duration of that image. When the image terminates, VAX/VMS restores the process's previous name.

VAX/VMS qualifies a process name by the group number of its user identification code (UIC). Therefore, process names need be unique only within the group.

2.1.4 Process Identification

VAX/VMS assigns every process and subprocess a process identification number that is unique throughout the system. The identification of a process does not change as long as the process remains in existence.

You must specify a process identification to affect processes outside your group.

VAX/VMS ENVIRONMENT

You can use the Show Process command to display your process name and identification, or you can use the Show System command to display the name and identification of other processes.

2.2 USER AUTHORIZATION FILE

The VAX/VMS system manager maintains a file called the user authorization file. It contains one entry for each user who is allowed access to the system. Each entry includes the following information:

- User name
- Password
- UIC
- User's account name
- User's default disk device
- User's default command interpreter
- Name of user's default directory
- Name of an indirect command file or command procedure to execute at login, for example LOGIN.CMD.
- Default file protection to be applied to newly-created files
- Privileges allowed this user
- Process priority

When you log in, VAX/VMS uses your authorization file entry to associate your UIC, privileges, and priority with the process it creates for you.

2.2.1 VAX/VMS UICs and Protection

Under VAX/VMS, a UIC is a unique 32-bit identification of a particular user. It consists of a 16-bit octal group number and a 16-bit octal member number, as illustrated in Figure 2-1.

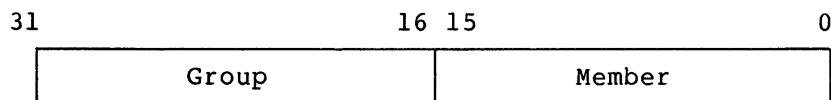


Figure 2-1 Format of VAX/VMS UICs

UIC group and member numbers range from 0 through 377 (octal).

VAX/VMS UICs are used to determine file and process protection.

2.2.1.1 File Protection - When you create a file, your UIC becomes the file-owner UIC. You specify read, write, execute, and delete file protection for system, owner, group, and world based on your UIC. This type of file protection is similar to file protection under RSX-11M.

Like protection under RSX-11M, VAX/VMS protection for each user category is represented by the four 4-bit fields RWED. The only difference in the interpretation of VAX/VMS and RSX-11M protection is in the E field. Under VAX/VMS, this field determines whether a user has the right to execute files that contain executable program images (either VAX-11 or RSX-11M images). When applied to an entire volume, the E field determines whether users can create files on the volume.

Under VAX/VMS, your UIC is not equivalent to your default directory name. That directory name is specified as a separate entry in the user authorization file. Section 2.3.2 discusses directory names in greater detail.

2.2.1.2 Process Protection - VAX/VMS also uses UICs for process protection. Process protection determines which processes you can affect using any of the following means:

- An MCR or DCL command that accepts a process name as a parameter, for example, the MCR commands Cancel and Stop
- An RSX-11M image that issues directives specifying another process as the target, for example, ABORT TASK
- A native image that issues system services specifying another process as the target, for example, the Suspend Process system service

A process always can affect the subprocesses that it creates.

In addition, using UIC-based process protection in conjunction with process privileges, VAX/VMS allows a process to affect other processes in the following categories:

- Processes having the same UIC group number as the requesting process
- All processes in the system regardless of UIC

2.2.2 Privileges

RSX-11M recognizes users as privileged or nonprivileged based on the group number of the UIC. Group numbers equal to or less than 10 (octal) are privileged. Once a user has logged in, RSX-11M associates the privileged or nonprivileged status with the terminal from which the user is issuing commands.

VAX/VMS differs from the RSX-11M approach to privileges in two respects:

- User privileges are associated with the process VAX/VMS creates for you, rather than being associated with your terminal.

VAX/VMS ENVIRONMENT

- Users are not categorized as privileged or nonprivileged; rather, the VAX/VMS system manager can associate a wide range of privileges with each user on an individual basis depending on each user's needs.

VAX/VMS controls the functions that you are allowed to request based on the privileges specified in your user authorization file entry. All of the privileges defined by VAX/VMS are listed in Appendix A. The privileges that an MCR user requires are normally the same as those that a DCL user requires.

If, during execution, an image attempts to use a function for which it does not have the privilege, it receives an error status.

The individual command descriptions in Chapter 4 indicate the privileges required for each command. If you attempt to use a command for which you do not have the appropriate privilege, the system issues an error message. Error messages are described in the VAX/VMS System Messages and Recovery Procedures Manual.

Two frequently referred to privileges are group process control privilege and world process control privilege. A process with group process control privilege can affect other processes that have the same UIC group number.

A process with world process control privilege can affect any process regardless of its UIC.

2.2.3 Process Priority

The user authorization file entry also specifies the priority at which your process executes. VAX/VMS priorities range from 1 through 15 for normal processes and from 16 through 31 for time-critical (real-time) priorities. In RSX-11M, a task has a priority assigned to it. When a user runs a task, the task executes at its priority. Under VAX/VMS, however, all images execute at the requesting user's priority.

2.3 FILE SPECIFICATIONS

RSX-11M and VAX/VMS file specifications are similar. The main areas of difference are in the device and directory fields, as described below. Differences in the device field are related to VAX/VMS's use of a letter to designate the controller. Differences in the directory field result from differences in Files-11 Structure Levels 1 and 2. VAX/VMS file specifications have the following format.

node::device:[directory]filename.type;ver

node Node name. The node name is a 1- to 6-character alphanumeric string that identifies a location on the network. The node name is separated from the device name by two colons (::).

VAX/VMS ENVIRONMENT

device Device name. Device names are specified in the following format.

 devcu:

dev is a mnemonic for the device type.
c is a controller designation.
u is the device unit.

The maximum length of the device name field is 15 characters.

directory Directory name. Directory names are specified in one of the following formats.

[g,m]
[name]
[name.name.name,,,]

Under VAX/VMS, all of the above formats can be used for structure level 1 and 2 disks; however, only the [g,m] format can be used under RSX-11M.

Angle brackets (<>) can be used instead of brackets ([]) to delimit the directory name.

filename File name. A file name consists of 1 to 9 alphanumeric characters.

type File type. A file type consists of 1 to 3 alphanumeric characters.

ver File version number. Version numbers are decimal numbers ranging from 1 through 32767. You can use either a semicolon (;) or a period (.) to separate a file type from a file version number.

File names, file types, and version numbers apply only to files on mass storage volumes. Directory names apply only to files on disk volumes. For record-oriented devices, only the device name field of the file specification is required.

You can use wild cards (*) in the file name, file type, and version number fields of a file specification.

2.3.1 Device Names

VAX/VMS device names consist of a device mnemonic, controller designation, and unit number.

2.3.1.1 Device Mnemonic - The device mnemonic can be any one of those listed in Table 2-1.

VAX/VMS ENVIRONMENT

Table 2-1
VAX/VMS Device Types

Mnemonic	Device Type
CR	Card reader
DB	RP04, RP05, and RP06 disk
DM	RK06 and RK07 disk
DX	Floppy disk
DR	RM03
LP	Line printer
MT	TE16 magnetic tape
NET	Network communication device
TT	Interactive terminal
XM	DMC11

Devices supported by both VAX/VMS and RSX-11M have the same mnemonics under both systems.

You need not load the driver to use any of the devices listed in Table 2-1.

2.3.1.2 Controller Designation and Unit Numbers - The device controller is designated by an alphabetic letter (A through Z). For example, MTA designates magnetic tape controller A.

Unit numbers are decimal numbers ranging from 0 through 65535. VAX/VMS unit numbers start at 0 for each controller; for example, a system can have two tape units on two different controllers designated as MTA0 and MTB0. Each is unit 0 on its controller.

2.3.2 Directory Names

Directory names are represented in the [g,m] format or by a 1- to 9-character alphanumeric string. Using subdirectories, you can specify up to seven directory levels after the main directory name in the following format.

[name.name.name,,,]

When subdirectories are used, the comma in a UFD-format directory or subdirectory name is omitted; leading zeros are required. An example of a directory name is [122020.MSG]. Directories are described further in the VAX/VMS Command Language User's Guide.

You must use the DCL command CREATE/DIRECTORY to create directories on a structure level 2 volume. That command is described in the VAX/VMS Command Language User's Guide. You can use the MCR UFD command to create structure level 1 directories.

Any volumes that are to be transported between VAX/VMS and RSX-11M systems must have directories in the [g,m] format; the comma can be omitted. For example, [123050] and [123,050] are equivalent on both systems.

2.3.3 Version Numbers

RSX-11M displays version numbers in octal. VAX/VMS displays them in decimal. However, both systems maintain version numbers in binary; version numbers are compatible internally. To determine version numbers when moving files between systems, use PIP on either VAX/VMS or RSX-11M to list file name and version information as it appears in that system. The PIP supplied with VAX/VMS has been modified to print version numbers in decimal. You can also use the Directory command under VAX/VMS to display the content of a directory.

2.3.4 Defaults in File Specification

When a field of a file specification is omitted, VAX/VMS supplies defaults. Table 2-2 lists the defaults for each field.

Table 2-2
File Specification Defaults

Field	Default
node	Local system
device	Disk established as your default by 1) your entry in the user authorization file, or 2) execution of a Set Default command. The system disk is not necessarily the default.
directory	Directory established as your default by 1) your entry in the user authorization file, or 2) execution of a Set Default or Set/UIC command.
file name	None
file type	File type that is the default for the particular command or component being used. Table 2-3 lists default file types.
file version	The rules for establishing default file version numbers are the same as those used in RSX-11M. For input files, the system assumes the most recent version; that is, the highest version number. For output files, the system increases the version number by 1 for existing files and supplies a version number of 1 for new files.

VAX/VMS ENVIRONMENT

Table 2-3
Default File Types

File Type	File Content
B2S	Source input for the PDP-11 BASIC-PLUS 2/VAX compiler
CBL	Source input to the PDP-11 COBOL-74/VAX compiler
CMD	RSX-11M indirect command file
COM	VAX/VMS command procedures
COR	AN SLP correction file
DAT	Input or output data file
DIR	Directory file
FTN	Source language input for the PDP-11 FORTRAN-IV compiler
EXE	Image created by the VAX-11 Linker or the VAX/VMS version of the RSX-11M task builder
FOR	Source input for the VAX-11 FORTRAN-IV-PLUS compiler
LIS	Listing file produced by a VAX-11 compiler or assembler
LOG	Batch job output file
LST	Listing file produced by a PDP-11 compiler or assembler
MAC	Source input to the PDP-11 MACRO-11 assembler
MAP	Memory allocation (map) listing produced by the VAX-11 Linker or the RSX-11M task builder
MAR	Source input to the VAX-11 MACRO assembler
MLB	VAX/VMS or RSX-11M macro library
OBJ	Object module produced by a PDP-11 or VAX-11 assembler or compiler
ODL	Overlay description input to RSX-11M task builder
OLB	VAX/VMS or RSX-11M object module library
OPT	Options file for input to the VAX-11 Linker
STB	Symbol table file created by the VAX-11 Linker or RSX-11M task builder

2.4 LOGICAL NAMES

The VAX/VMS logical name capability is an extension of that available under RSX-11M. Using the VAX/VMS Assign command, you can assign a logical name to either of the following:

- A physical device, logical device, or pseudo-device name
- A file specification or portion of one

Section 3.2.3 describes the uses of logical names under VAX/VMS.

VAX/VMS categorizes logical names differently from RSX-11M. Rather than categorizing them as login, local, and global logical names, VAX/VMS describes them as process-local, available within the group, and available to all processes. VAX/VMS maintains a logical name table for each category:

- Process logical name table
- Group logical name table
- System logical name table

The Show Logical command displays the logical names and their equivalences from the three tables.

When VAX/VMS encounters a logical name in a command or program, it translates the logical name to its equivalence name. To do so, it searches the process, group, and system tables, in that order, and uses the first match it finds. That is, entries in the process table take precedence over those in the group and system tables, and entries in the group table take precedence over those in the system table. Once VAX/VMS has translated a logical name, it applies any appropriate file specification defaults.

2.4.1 Process Logical Name Table

The process logical name table contains the logical names that are local to the process. Once an entry is assigned to the process logical name table, it is available to all images that run in the process until it is deassigned, or until the process is deleted (for example, as a result of log out).

2.4.2 Group Logical Name Table

The group logical name table contains the logical names that are available to all processes that have the same group number in their UICs as the process that created the entry. Unlike RSX-11M, VAX/VMS does not use the identification of the initiating terminal to determine whether a process has access to the group logical name table. Entries remain in this table until explicitly deleted.

User privilege is required to place and delete a name in the group logical name table.

2.4.3 System Logical Name Table

The system logical name table contains entries available to all processes in the system. The entries in it are used in a manner similar to the way global assignments are used under RSX-11M. Entries remain in the table until explicitly deleted.

User privilege is required to place and delete a name in the system logical name table.

2.4.4 System-Defined Logical Names

VAX/VMS defines logical names at the process and system levels. The system assigns logical names for every process created during login or for execution of a batch job:

- **SYS\$INPUT** -- command input stream for the process. For an interactive user, **SYS\$INPUT** is equated to the terminal. For a batch job, **SYS\$INPUT** is equated to the batch input stream.
- **SYS\$OUTPUT** -- command output stream for the process. For an interactive user, **SYS\$OUTPUT** is equated to the terminal. For a batch job, **SYS\$OUTPUT** is equated to the batch job log file. For indirect command files, you can temporarily redirect **SYS\$OUTPUT** to a different file as described in Section 4.15.
- **SYS\$COMMAND** -- original **SYS\$INPUT** for a job. When a process is executing an indirect command file, **SYS\$INPUT** is assigned to that file; **SYS\$COMMAND** remains assigned to the original command stream.
- **SYS\$ERROR** -- default output stream to which the system writes messages. For an interactive user, **SYS\$ERROR** is equated to the terminal. For a batch job, **SYS\$ERROR** is equated to the batch job log file.
- **SYS\$DISK** -- default disk device established at log in, or established by the Set Default command.

The logical names listed above are in the process logical name table.

VAX/VMS defines the following entries for the system logical name table:

- **SYS\$SYSTEM** -- device and directory of the system disk that contains the operating system program and procedures.
- **SYS\$LIBRARY** -- device and directory that contains system libraries.

In addition, the system manager at your installation can place names in the system logical name table.

VAX/VMS ENVIRONMENT

2.5 DEVICE MAPPING

Device mapping is the technique of equating an RSX-11M device name with a VAX/VMS device name. VAX/VMS automatically performs device mapping for two categories of device names:

- RSX-11M pseudo-device names, for example, TI
- RSX-11M physical device names, for example, DB2

Mapping occurs when an image uses an RSX-11M device name, or when a command issued either interactively or from an indirect command file uses an RSX-11M device name.

2.5.1 Mapping RSX-11M Pseudo-Device Names

VAX/VMS maps RSX-11M pseudo-device names to VAX/VMS logical names that serve similar functions. Table 2-4 shows the correspondence between RSX-11M pseudo-device names and VAX/VMS logical names.

Table 2-4
Mapping of RSX-11M Pseudo-Device Names

RSX-11M Pseudo-Device Name	VAX/VMS Logical Name
TI (for input)	SYS\$INPUT
TI (for output)	SYS\$OUTPUT
CO	SYS\$COMMAND
CL	SYS\$ERROR
SY	SYS\$DISK
LB	Device assigned by VAX/VMS system manager
WK	Device assigned by VAX/VMS system manager
SP	Device assigned by VAX/VMS system manager
OV	Device containing image file

Devices TI, CO, and CL have permanently open files associated with them. An RSX-11M image interacts with the process-permanent files by issuing I/O requests on TI, CO, and CL that VAX/VMS automatically converts to requests on SYS\$INPUT, SYS\$OUTPUT, SYS\$COMMAND, and SYS\$ERROR.

2.5.2 Mapping RSX-11M Physical Device Names

You can control the association of an RSX-11M physical device name with an actual VAX/VMS device, or VAX/VMS can perform the association automatically. To associate an RSX-11M device name with a specific VAX/VMS device unit, issue an Assign or Mount command that specifies the RSX-11M device name as the logical name for the device, as described in Chapter 3.

If you do not use logical names, VAX/VMS automatically maps the RSX-11M device name to a VAX/VMS device name. VAX/VMS performs the mapping by retaining the device type (for example, MT) and converting the RSX-11M unit number into the corresponding VAX/VMS controller letter and unit number.

VAX/VMS performs the mapping to a physical device by converting the RSX-11M unit number to decimal and dividing by 16 (decimal). The quotient is added to the ASCII value representing the character A (65). The result is the controller letter. The remainder becomes the VAX/VMS unit number. For example, RSX-11M devices TT0 and DB22 become VAX/VMS devices TTA0 and DBB2, respectively, as follows.

TT0 to TTA0:

$$\text{Controller and unit} = A + \frac{0}{16} = A + 0 \text{ with a remainder of } 0$$

$$\begin{aligned} A + 0 &= 65 = \text{controller} \\ 0 &= \text{unit number} \end{aligned}$$

DB22 to DBB2:

$$\text{Controller and unit} = A + \frac{18}{16} = A + 1 \text{ with a remainder of } 2$$

$$\begin{aligned} A + 1 &= 66 = B = \text{controller} \\ 2 &= \text{unit number} \end{aligned}$$

VAX/VMS performs this conversion when assigning an I/O device for an RSX-11M image.

2.6 TERMINAL FUNCTIONS

Table 2-5 lists the terminal control key sequences that VAX/VMS recognizes. With the exception of CTRL/Y and CTRL/X, these terminal functions are essentially the same as those available under RSX-11M. CTRL/Y always causes a return to the command interpreter. CTRL/X cancels the current line and deletes data in the type-ahead buffer. Under VAX/VMS, you do not have to wait for one command or program to terminate before typing the next request. Unprocessed commands are stored in the type-ahead buffer.

VAX/VMS ENVIRONMENT

Table 2-5
Terminal Function Keys

RETURN	(Carriage return) Transmits the current line to the system for processing.
Control characters	Before a terminal session, initiates login sequence. Define functions to be performed when the CTRL key and another key are pressed simultaneously. All CTRL/x key sequences are echoed on the terminal as ^x.
CTRL/C	Before a terminal session, initiates login sequence. During command entry, cancels command processing. Certain system and user programs provide special routines to handle CTRL/C interrupts. If CTRL/C is pressed to interrupt a program that does not handle CTRL/C, CTRL/C has the same effect as CTRL/Y and echoes as ^Y.
CTRL/I	Duplicates the function of the TAB key.
CTRL/K	Advances the current line to the next vertical tab stop.
CTRL/L	Requests form feed.
CTRL/O	Alternately suppresses and continues display of data at the terminal.
CTRL/Q	Restarts terminal output that was suspended by CTRL/S.
CTRL/R	Retypes the current line during input and leaves the cursor positioned at the end of the line.
CTRL/S	Suspends terminal output until you press CTRL/Q.
CTRL/U	Cancels the current line and discards it.
CTRL/Y	Interrupts command or program execution and returns control to the command interpreter.
CTRL/X	Cancels current line and deletes data in the type-ahead buffer.
CTRL/Z	Signals end-of-file for data entered from the terminal.

(continued on next page)

VAX/VMS ENVIRONMENT

Table 2-5 (Cont.)
Terminal Function Keys

TAB	Moves the printing element or cursor on the terminal to the next tab stop on the terminal. Most terminals have tab stops at every 8 character positions on a line.
DELETE	Deletes the last character entered at the terminal and backspaces over it. On some terminals, this key is labeled RUBOUT.
ESCAPE	Has special uses to particular commands or programs, but generally performs the same function as RETURN. On some terminals, this key is labeled ALTMODE.

2.7 SYNTAX RULES

VAX/VMS MCR commands have the following general syntax:

```
command[/cmd-keywords] parameter[/keywords...][...]
```

The rules for separating items on a VAX/VMS command string are the same as those for RSX-11M, that is:

- At least one space must separate the command from the first parameter, and at least one space must separate each additional parameter from the previous parameter. Multiple blanks and tabs are permitted in all cases where a single blank is required.
- Each keyword must be preceded by a slash (/). Any number of spaces, including zero, can precede the slash.

2.7.1 Continuing Commands on More than One Line

You can enter a command string on more than one line by using the continuation character, a hyphen (-), as the last element on a line, as in the following example.

```
> COPY /LOG -  
>_   OUTFIL.DAT /CONTIGUOUS = -  
>_   INFIL1.DAT,INFIL2.DAT
```

No restriction is placed on the number of lines used to enter a command string.

After each continued line, the command interpreter prompts for additional input (>_).

2.7.2 Commenting Command Lines

VAX/VMS MCR distinguishes command lines from comments using the same conventions as RSX-11M uses, that is:

- A semicolon (;) as the first significant character on a line indicates that the entire line is a comment.
- An exclamation mark (!) indicates the beginning of a comment in a command line.

2.7.3 Truncating Command Names and Keywords

You can truncate any item in a command that appears in uppercase letters in the individual command descriptions in Chapter 4.

All command names can be truncated to their first three characters. You can truncate command names to fewer than three characters as long as the name remains unique. For example, because the Type command is the only command beginning with the letters TY, it has a minimum truncation of two letters. For convenience, the following frequently used commands can be truncated to one letter even though other commands start with the same letter:

- Continue
- Deposit
- Examine
- Run

All other portions of a command line (for example, keywords and keyword values) can be truncated to four letters, or fewer if they remain unique within the command.

2.7.4 Placement of Keywords

VAX/VMS MCR is considerably more flexible than RSX-11M MCR about where you can place keywords in the command string. The command descriptions in Chapter 4 separate command keywords from file specification keywords. Command keywords have the same meaning regardless of whether they appear following the command name or following a command parameter. For example, the following two commands are identical.

```
> RUN /UIC=[200,230] DOTS.TSK
> RUN DOTS.TSK /UIC=[200,230]
```

File specification keywords, on the other hand, have different meanings depending on where they are placed in the command string. If specified immediately following a file specification, they affect only the file thus qualified. If specified following the command name, they affect all files specified as parameters. For example, the first command below results in the printing of two copies of both files,

VAX/VMS ENVIRONMENT

whereas the second results in the printing of two copies of SANDPIPER.MAP and one copy of YELLOWLEG.MAP.

```
> PRINT/COPIES=2 SANDPIPER.MAP, YELLOWLEG.MAP
> PRINT SANDPIPER.MAP/COPIES=2, YELLOWLEG.MAP
```

Some file specification keywords are not valid as command keywords. Individual keyword descriptions indicate any restrictions.

2.7.5 Specifying File Protection

All disk and tape volumes have protection codes that restrict access to the volume. This protection is assigned when the volume is initialized, and can be reassigned when a volume is mounted.

For disk volumes, each file on the volume, including a directory file, can have a different protection associated with it.

Files potentially can be accessed by users in four categories:

- SYSTEM -- all users who have group numbers of 1 through 10 (octal) and users with physical I/O or logical I/O privilege (generally, system managers, system programmers, and operators)
- OWNER -- the UIC of the person who created and, therefore, owns the volume or file
- GROUP -- all users who have the same group number in their UICs as the owner of the file
- WORLD -- all users who do not fall into any of the other three categories

Each of these categories of user can be allowed or denied the following types of access:

- READ -- the right to examine, print, or copy a file or files on a volume
- WRITE -- the right to modify the file or to write files on a volume
- EXECUTE -- the right to execute files that contain executable program images (when applying protection to an entire volume, this field is interpreted as the right to create files on the volume)
- DELETE -- the right to delete the file or files on the volume

Any combination of access types can be specified for any category of user.

When you specify a protection code, you must abbreviate protection types to one character; you can specify the user categories and protection types in any order. If you omit a protection type for a user category, that category of user is denied that type of access. When specifying file protection, if you omit a user category, the current access rights for that category remain unchanged. When you specify volume protection, omission of a category denies that user category all access.

VAX/VMS ENVIRONMENT

When you specify a protection code, separate each user category from access type with a colon. If you wish to deny all access to a category, specify the category without the trailing colon, for example, GR. To specify more than one user category, separate the categories with commas and enclose the entire code in parentheses, as follows.

```
SET PROTECTION=(SYS:RWED,GR:R,W)/DEFAULT
```

This protection code allows the system all types of access; allows group members read access only; prohibits all access by users in the world category; and does not change the current default for the file's owner.

2.7.6 Entering Dates and Times

When a command accepts a keyword that specifies a time value, the time value is either an absolute time or a delta time:

- An absolute time is a specific date and time of day, for example, 10-JUN-1978 10:53:22.10.
- A delta time is a future offset from the current date and time of day, for example, 2 days and 3 hours from now.

2.7.6.1 Absolute Times -- Absolute times have the following format.

```
[dd-mmm-yyyy[:]][hh:mm:ss.ss]
```

You can specify either the date or the time, or both. The variable fields are as follows:

<u>Field</u>	<u>Meaning</u>
dd	Day of month (1 through 31)
mmm	Month; the month must be specified as one of the following 3-character abbreviations: JAN, FEB, MAR, APR, MAY, JUN, JUL, AUG, SEP, OCT, NOV, DEC
yyyy	Year
hh	Hour of the day (0 through 23)
mm	Minute of the hour (0 through 59)
ss.ss	Seconds and hundredths of seconds (00.00 through 59.99)

The punctuation marks indicate how the system interprets the time value you enter, as follows:

- If you specify both the date (dd-mmm-yyyy) and the time (hh:mm:ss.ss), you must type the colon between the date and the time.
- You can truncate the date and the time on the right; however, if you are specifying both a date and a time, the date part must contain at least one hyphen.

VAX/VMS ENVIRONMENT

- You can omit any of the fields within the date or time, as long as you type the punctuation marks; the system supplies default values.

When you omit the date or any of its fields from an absolute time value, the system supplies the current day, month, and year by default.

When you omit any fields from the time, the system supplies a value of 0 for the field.

Examples:

<u>Time Specification</u>	<u>Result</u>
28-JUN-1978:12	12:00 noon on June 28, 1978
28-JUN	Midnight (00:00 o'clock) on the 28th of June, this year
15	3:00 p.m., today
15-	The 15th day of the current month and year, at midnight
18:30	6:30 p.m., today
15--::30	00:30 o'clock, on the 15th day of the current month

When you specify an absolute time that has already passed, the system performs the request immediately.

2.7.6.2 Delta Times -- Delta times have the format:

[dd-][hh:mm:ss.ss]

The variable fields are as follows:

<u>Field</u>	<u>Meaning</u>
dd	Number of days, 24-hour units (0 through 99)
hh	Number of hours (0 through 23)
mm	Number of minutes (0 through 59)
ss.ss	Number of seconds and hundredths of seconds (00.0 through 59.99)

When you specify a delta time value, you can truncate the time field on the right; you may also omit any of the variable fields, as long as you supply the punctuation marks. When any field is omitted from a delta time value, the system supplies a value of 0 for the field.

Examples:

<u>Time Specification</u>	<u>Result</u>
3-	3 days from now (72 hours)
3	3 hours from now
:30	30 minutes from now
3-:30	3 days and 30 minutes from now
15:30	15 hours and 30 minutes from now

2.8 ERROR MESSAGES

Under VAX/VMS, when you enter an MCR command incorrectly, the command interpreter issues a message in the standard VAX/VMS format. These messages do not look like RSX-11M MCR error messages. Rather, they have the following general format.

%XXX-L-CODE, text

XXX is a mnemonic for the operating system program issuing the message.

L is a severity level indicator: S for success; I for information; W for warning; E for error; and F for fatal.

CODE is a shorthand code for the message text that follows the code.

For example:

%MCR-W-MAXPARM, maximum parameter count exceeded

Because VAX/VMS messages are descriptive, you can usually learn the corrective action from them. The VAX/VMS System Messages and Recovery Procedures Manual lists all system error messages and describes corrective measures.

RSX-11M components (for example, PIP and TKB) issue the same messages under VAX/VMS as they do under RSX-11M. When using VAX/VMS MCR to run RSX-11M components, you should consult the appropriate RSX-11M Version 3.1 documentation for error message information.

CHAPTER 3

USING VAX/VMS MCR

Using the VAX/VMS MCR interface, you can perform four types of work:

- Run RSX-11M images and VAX-11 images
- Use RSX-11M components for RSX-11M program development, for example, MACRO-11 or the task builder
- Use VAX/VMS components for native program development, for example, VAX-11 MACRO and the linker
- Execute RSX-11M indirect command files, for example, RSX-11M/S system generation indirect command files

This chapter explains how to log into the VAX/VMS system and perform the types of work listed above. In addition, it describes considerations for task building RSX-11M images that are to run under VAX/VMS.

3.1 LOGGING INTO VAX/VMS

To log into the VAX/VMS system, you must first gain the attention of login. To do this, press one of the following control sequences:

- CTRL/C
- CTRL/Y
- RETURN

Login prompts for your user name, and then for your password:

Username:

Password:

When you enter your password, the system does not display it.

Using your entry in the user authorization file, login verifies your right to gain access to the system and sets up the default characteristics for the terminal session. It then executes the login indirect command file specified in your user authorization file entry, if any. If one is not specified, login searches your default directory for an indirect command file named LOGIN.CMD and executes the file if one is present. Finally, a command interpreter prompts for command input.

3.1.1 Selecting a Command Interpreter

The system determines which command interpreter is to be associated with your process using the following factors:

- The default command interpreter named in your user authorization file entry
- The /CLI keyword you specified following your user name to override the default command interpreter

If the MCR command interpreter is not your default and you wish to use it, enter your user name followed by /CLI=MCR, as follows.

Username: CALEB/CLI=MCR

Then, enter your password in the normal manner.

3.1.2 LOGIN.CMD File

The LOGIN.CMD file used under VAX/VMS is identical in function to a LOGIN.CMD file under RSX-11M. If your default command interpreter is MCR, or if you request MCR using the /CLI keyword to login, the MCR command interpreter searches your default directory for the file LOGIN.CMD. LOGIN.CMD must contain valid VAX/VMS MCR commands and directives.

When you log in using the DCL command interpreter, it looks for a LOGIN.COM command procedure, as described in the VAX/VMS Command Language User's Guide.

3.2 PREPARING TO RUN AN IMAGE

The basic steps required to prepare an image for execution are the same for VAX-11 and RSX-11M images:

- Allocate and mount any physical devices that the image requires
- Equate VAX/VMS physical device names to any logical device names or RSX-11M physical device names that the image uses

To execute, the image must reside on a Files-11 disk volume. The volume must be mounted and you must have the right to gain access to it; that is, you must be allowed execute (E) access.

3.2.1 Allocating Devices

Device allocation under VAX/VMS is essentially the same as device allocation under RSX-11M; that is, you use the Allocate command to reserve a device. Once allocated, the device is not available to other users. The device remains allocated until you deallocate it or log out of the system. Under RSX-11M, allocated devices are reserved for use by any tasks that you initiate. Under VAX/VMS, on the other hand, allocated devices are reserved for any images that run in your current process or one of its subprocesses.

The VAX/VMS Allocate command accepts as a parameter either a logical device name or a VAX/VMS physical device name. If you specify a physical device name, it can be either explicit or generic.

3.2.1.1 Generic Device Names - A generic device name is one that explicitly specifies the device type, and optionally, the controller or unit. A generic device name allows the system to select the actual device unit to be allocated from available devices of the specified type. For example, the following command requests VAX/VMS to allocate any available TEL6 tape drive.

```
> ALL MT:
  _MTB2: ALLOCATED
```

The system response indicates that drive 2 on controller B has been allocated. When the device type and controller are specified in a generic device name, the Allocate command looks for an available unit on the specified controller. Similarly, when the device type and unit number are specified, the Allocate command looks for a controller of the designated type that has the specified device unit available.

Unlike RSX-11M, when you omit the controller and unit number, VAX/VMS does not default to unit 0 of controller A. This difference may affect indirect command files that allocate devices expecting unit 0 to be used by default; see Chapter 5.

Chapter 3 of the VAX/VMS Command Language User's Guide provides additional information about allocating tape and disk units.

3.2.2 Mounting Volumes

Volumes are mounted in essentially the same way under VAX/VMS as they are under RSX-11M; both systems support a Mount command. Both systems define similar functions requested by specifying Mount command keywords. The actual keyword names, however, differ between systems. The VAX/VMS Mount command also has additional options that function as follows:

- Define a logical name for the device on which the volume is mounted
- Specify volume owner and volume protection
- Specify the degree of sharing that is allowed on the volume, for example, specify a private volume or one that is available to the group or world

Section 4.21 describes all of the Mount command options.

The main purpose of the Mount command is to gain access to the volume. Under VAX/VMS, however, you also can use certain disk volumes without issuing a Mount command for them yourself. Any volume that another user (most likely the system manager or operator) has mounted specifying the /SYSTEM keyword is available to anyone logged into the system. Any volume that another user has mounted specifying the /GROUP keyword is available to all users with the same UIC group number as the user who issued the Mount command.

The Mount command accepts either a logical device name or a physical device name as a parameter. If you specify a physical device name, it

can be either explicit or use defaults. The default for the controller designator is A, and the default for unit number is 0.

Chapter 3 of the VAX/VMS Command Language User's Guide provides additional information about mounting disk and tape volumes.

3.2.3 Assigning Logical Names

Under VAX/VMS, you can use logical names in RSX-11M images, native images, and MCR commands. Using the Assign command, you can equate a logical name with a VAX/VMS physical device name, as in the following example:

```
ASN DBB2:=IN0:
```

Or, you can equate a logical name with all or a portion of a file specification, as in the following example.

```
ASN DBB2:[MYDIR]FILEA,DAT;2=INPUT
```

Unlike RSX-11M, VAX/VMS logical names are not restricted to two letters and a unit number; VAX/VMS logical names can be up to 63 alphanumeric characters. The following sections describe the use of logical names for RSX-11M images and native images.

3.2.3.1 Using Logical Names for RSX-11M Images - RSX-11M images issue ASSIGN LUN directives to assign a logical unit number (LUN) to an RSX-11M physical device unit or a logical name. Similarly, the RSX-11M task builder ASG option can be used to assign a LUN to a physical or logical device at build time.

When either method is used to assign a LUN to an RSX-11M physical device unit, you can use the Assign command to define the RSX-11M physical device name as the logical name for a VAX/VMS device. Then, when VAX/VMS encounters the RSX-11M name in the program, it translates the name to the associated VAX/VMS physical device name. The following example illustrates the use of the Assign command.

```
ASN DBA1:=DK2:
```

When the ASSIGN LUN directive is issued or when the image activated had a task builder option to assign a LUN to a logical device name, you can use the Assign command to equate the logical name with a VAX/VMS physical device name. For example:

```
ASN DBB0:=XY:
```

If you do not assign logical names, VAX/VMS attempts to map the RSX-11M device names used by the image to VAX/VMS device names, as described in Section 2.5.2.

3.2.3.2 Using Logical Names Instead of Reassign - Under RSX-11M, a privileged user can assign a physical or logical device name to a LUN using the Reassign (REA) command.

Under VAX/VMS, you must assign LUNs at either the program or task build level; the VAX/VMS MCR command interpreter does not support the Reassign command. However, the logical name facility provides the

flexibility needed to associate the RSX-11M device name used in an ASSIGN LUN directive or ASG option with a VAX/VMS physical device, as described above.

3.2.3.3 Using Logical Names Instead of Redirect - Under RSX-11M, a privileged user can redirect I/O operations from one device to another during task execution. VAX/VMS does not support the Redirect (RED) command; you must assign logical names prior to running the image.

3.2.3.4 Using Logical Names for Native Images - The VAX/VMS logical name capability provides more flexibility than the RSX-11M approach. Native images can use a logical name that is equated with all or part of a file specification starting with the leftmost item of the file specification. Similarly, you can use logical names that are equated to file specifications at the command level. The following is an example of using logical names.

```
> ASN DBB1:[JONES]SNOW.DAT;4=INPUT
> RUN ANLZE
```

```

.
.
.
```

The Assign command equates the logical name INPUT with the full file specification DBB1:[JONES]SNOW.DAT;4. When the image ANLZE refers to INPUT, VAX/VMS translates it to the associated file specification.

The full VAX/VMS logical name capability is available through VAX/VMS MCR; the VAX/VMS Command Language User's Guide provides more detailed information.

3.2.4 Installing RSX-11M Images

Under VAX/VMS, images are not installed before execution. You run images under VAX/VMS in the same manner that a nonprivileged user runs them under RSX-11M; that is, by supplying the file specification of the image as a parameter to the run command. VAX/VMS does not support the RSX-11M Install command.

You can use logical names to create an effect similar to Install. Assigning a logical name to the file specification of the image file allows the logical name to serve as a task name; typing the logical name in response to the MCR prompt (>) causes VAX/VMS to perform logical name translation and run the associated image.

For example, the following command establishes REP as the logical name for the image file DB4:[IMAGES]ALPHA.TSK contained on DBA4.

```
ASN DB4:[IMAGES]ALPHA.TSK=REP/GBL
```

Because the logical name REP is placed in the system logical name table, any user can type REP to run the image, as follows.

```
REP additional-data
```

The image can obtain the additional data by issuing a GET MCR COMMAND LINE directive just as it does in RSX-11M.

3.3 RUNNING IMAGES

The VAX/VMS command interpreter lets you run VAX/VMS and RSX-11M images. The basic format of the command to request image execution is as follows:

```
RUN file-spec
```

The parameter `file-spec` represents a standard VAX/VMS file specification or a logical name that translates to one. If the file type is not supplied, it defaults to `EXE`. To run RSX-11M images having a file type of `TSK`, you must either explicitly supply the file type, or you must rename the image files. `EXE` is the default file type for image files produced by the VAX-11 linker and the VAX/VMS version of the RSX-11M task builder.

The Run command provides keywords that control whether the requested image is run in your process, a subprocess, or a detached process. Additional keywords allow you to control the process's priority, scheduling, resource limits, and privileges. Section 4.25 describes all of the Run command options.

The following are examples of the Run command.

```
> RUN INVENTORY
```

The command above runs the image in the highest version of the file `INVENTORY.EXE` located on the default disk device and directory.

```
> ASN DBB2:[RSX] IMAGE.TSK;8=MYTASK
> RUN MYTASK
```

The sequence above assigns `MYTASK` as the logical name for the image. The Run command translates `MYTASK` to `DBB2:[RSX] IMAGE.TSK;8` and runs `IMAGE.TSK;8`.

Instead of using the Run command, you can run images in your process by entering the file specification of the image file in response to the MCR prompt (`>`), or by entering a logical name that translates to a file specification.

3.4 PROGRAM DEVELOPMENT UNDER VAX/VMS

VAX/VMS MCR permits the development of both VAX/VMS and RSX-11M images. Figure 3-1 illustrates some of the equivalent VAX/VMS and RSX-11M functions that you can perform using VAX/VMS MCR.

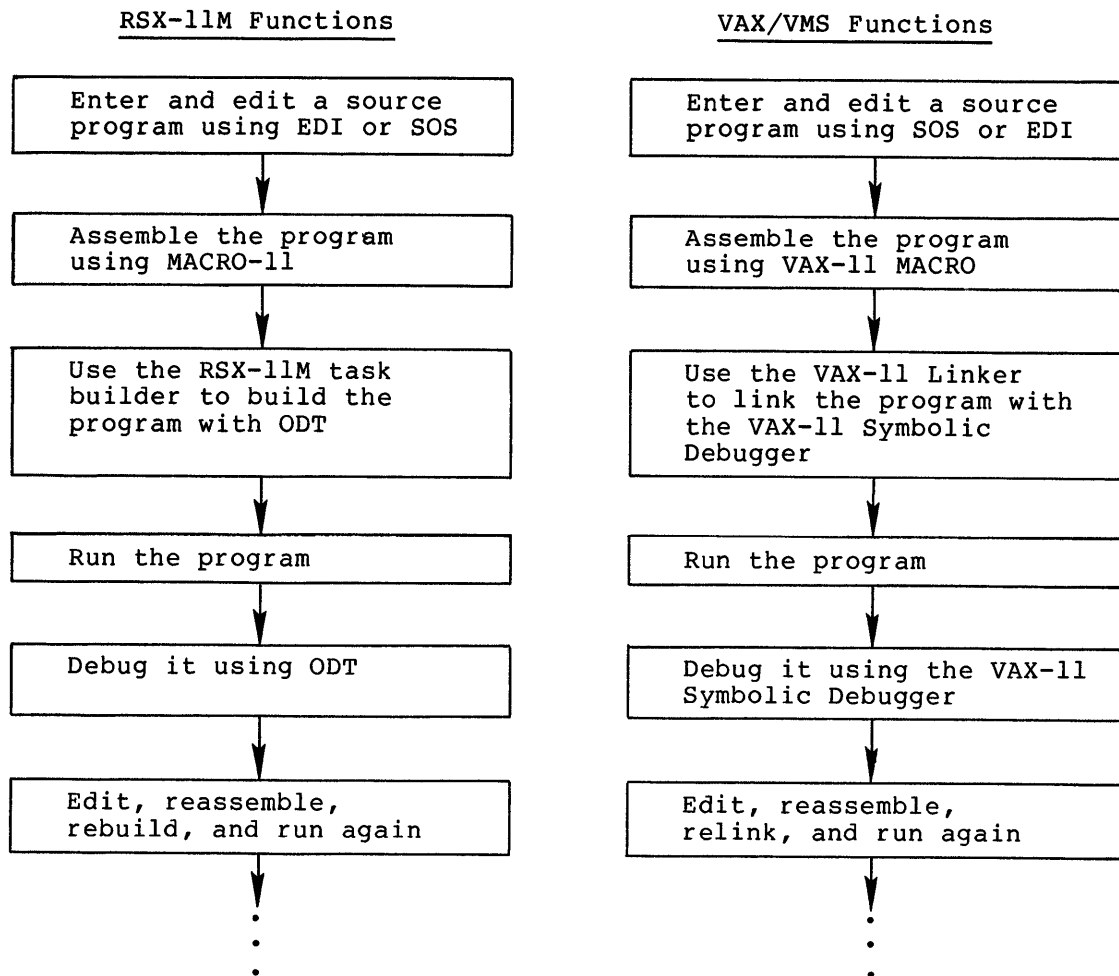


Figure 3-1 RSX-11M and VAX/VMS Program Development Cycles Using VAX/VMS MCR

3.4.1 Using RSX-11M Components

You can use any of the RSX-11M components listed in Section 1.2.1 by issuing a request to the VAX/VMS MCR command interpreter. To request any of these components, either enter the component name and press RETURN, or enter the component name followed by a valid command line for that component and press RETURN. Subsequent operation of the component is identical to that under RSX-11M, as documented in the related RSX-11M manuals.

In addition, you can build ODT into any RSX-11M image that requires debugging. ODT operates just as it does under RSX-11M. You also can interrupt image execution and type the Debug command to pass control to ODT; however, the image must have been built with ODT.

USING VAX/VMS MCR

Of the RSX-11M components available, all accept the standard VAX/VMS file specifications except VFY1 and FLX. Considerations for using FLX are provided in Section 3.4.2. The VAX/VMS Operator's Guide describes operation of VFY1.

VAX/VMS does not support the PIP/UF switch; you must use the UFD command, documented in Chapter 4, instead.

The following are examples of invoking RSX-11M components from VAX/VMS MCR.

```
> EDI
EDI> DBA1:[COLD]CENTIGRADE.DAT
[CREATING NEW FILE]
INPUT
.
.
.
<CR>
*EX
>
```

Typing EDI in response to the MCR prompt (>) and pressing RETURN causes EDI to prompt for a command line. EDI operates just as it does under RSX-11M. That is, the default directory is searched for the file CENTIGRADE.DAT. Failure to locate the file results in creation of a new file under directory [COLD]. Exiting from EDI causes MCR to prompt.

```
> MAC @MACINP
>
```

The command above requests the MACRO-11 assembler to read the indirect command file MACINP.CMD for command input. After the indirect file is processed, MCR prompts again.

```
> TKB APPLES=GRAPES
>
```

Typing TKB followed by a command line results in the task building of the object file GRAPES.OBJ on the default disk and directory to produce the image file APPLES.EXE, which is also stored on the default disk and directory. GRAPES.OBJ must have been produced by an RSX-11M compiler or assembler. When task building completes, VAX/VMS MCR prompts again.

```
> TKB
TKB> AVERAGE,AVERAGE=AVERAGE
TKB>/
ENTER OPTIONS:
TKB>
.
.
.
TKB>//
>
```

The sequence above invokes the RSX-11M task builder, supplies it with the names for the input and output files, and requests the task builder to prompt for options. The two slashes cause the task builder to exit; MCR prompts.

USING VAX/VMS MCR

```
> PIP DBB2:[GAULT]*.LST;*/DE
>
```

The command above invokes PIP to delete all listing (LST) files from directory GAULT on DBB2. VAX/VMS MCR prompts after the files are deleted.

```
> PIP
PIP> SPICE.*/PU
PIP> .
.
PIP> ^Y
>
```

In the above sequence, typing PIP in response to the MCR prompt causes PIP to prompt. PIP reprompts after each command until pressing CTRL/Y passes control to the MCR command interpreter.

3.4.2 Using FLX

FLX interprets only RSX-11M file specifications, that is, file specifications with the following format:

ddu:[g,m]filename.type;version

It cannot interpret a controller designation in the device name or a directory name (as opposed to the [g,m] form) in the file specification; nor does it accept a logical name for a file specification.

If you omit the directory from a file specification for a disk file, VAX/VMS uses your default directory. It can be in either the [g,m] form or a directory name.

Because FLX does not accept VAX/VMS device names, your use of FLX depends on the device mapping that the operating system performs automatically. Device mapping is described in Section 2.5.2.

If you specify a device name that does not map to a physical device on VAX/VMS, FLX issues an error message indicating that it could not find the device.

Before using FLX with a magnetic tape, you must allocate the tape drive.

The following are examples of FLX command lines.

```
> FLX DB0:[120,30]=MT17:SYS1.MAC/RT
```

VAX/VMS maps the device name DB0 to physical device DBA0. It maps device name MT17 to device MTB1.

```
> FLX
FLX> DR3:=DM0:[133,10]MYFIL.MAC/DO
```

VAX/VMS maps device name DR3 to physical device DRA3 and DM0 to DMA0. The user's default directory on DRA3 contains the output file.

3.4.3 Using VAX/VMS Components

From the VAX/VMS MCR command interface, you can also invoke the following native VAX/VMS components:

- VAX-11 MACRO (MAR)
- Standard VAX/VMS editor (SOS)
- VAX/VMS Librarian (LIB)
- VAX-11 Linker (Link command)
- Various VAX/VMS utilities available through the MCR command language (for example, Print, Submit, Set, and Show)

To request any of these components, type one of the following:

- The component name or the component name followed by a valid command line for that component
- The command that invokes the particular component

The information in parentheses in the list above indicates the appropriate means of requesting each component.

Subsequent operation of the requested component is described in the appropriate VAX/VMS document; see the VAX-11 Information Directory to determine the documents required.

3.5 EXECUTING RSX-11M INDIRECT COMMAND FILES

RSX-11M indirect command files have the same uses under VAX/VMS as under RSX-11M. That is, you can specify an indirect command file for processing by VAX/VMS MCR or you can specify an indirect command file in a command to an RSX-11M component executing under VAX/VMS. In either case, an at sign (@) precedes the file specification of the indirect file just as it does under RSX-11M.

Indirect files specified for use by RSX-11M components (for example, task builder and MACRO-11) are identical to those used under RSX-11M. Chapter 5 describes the MCR indirect file processor directives available under VAX/VMS and provides additional information about running MCR indirect command files. Section 4.15 describes the Execute procedure (@).

3.6 BUILDING RSX-11M IMAGES FOR EXECUTION UNDER VAX/VMS

The VAX/VMS system can serve as the host for 1) modifying existing RSX-11M images so that they execute under VAX/VMS or 2) developing new images to run under VAX/VMS or RSX-11M. When the image being modified or developed is to execute under VAX/VMS, you should build it with that fact in mind. The following sections describe considerations for building RSX-11M images to execute under VAX/VMS.

3.6.1 Task Builder Switches

You can use most of the switches defined for the RSX-11M task builder when building an image for execution under VAX/VMS. The only switch you cannot use is -MM (unmapped system); that is, the image must be built to execute on a mapped system.

Table 3-1 lists the switches that are effective under VAX/VMS. With the exception of -MM, unlisted switches have no effect under VAX/VMS and do not cause an error when the image executes under VAX/VMS. As a general rule, switches that are intended to affect the way RSX-11M treats the image during execution are ignored, for example, whether the image is checkpointable.

Table 3-1
Task Builder Switches Valid for RSX-11M Images under VAX/VMS

Switch or Option	Function
CC	Input file consists of concatenated object modules
DA	Image contains a debugging aid
DL	Specified library is a replacement for the system object module library
FP	Task uses the floating point processor (FPP only)
FU	All co-tree overlay segments are searched for matching definition or reference when modules from the default object module library are being processed
HD	Image includes a header
LB	Input file is a library file
MA	Memory allocation output includes information from the file
MM	System has memory management; for an image to run under VAX/VMS, MM must be present either explicitly or by default
MP	Input file contains an overlay description
MU	Image is separated into shareable and nonshareable portions
PI	Image is position independent
PM	Post-mortem dump is requested
SH	Short memory allocation file is requested
SP	Spool map output

(continued on next page)

USING VAX/VMS MCR

Table 3-1 (Cont.)
Task Builder Switches Valid for RSX-11M Images under VAX/VMS

Switch or Option	Function
SQ	Image program-sections are allocated sequentially
SS	Selective search for global symbols
TR	Image is to be traced
WI	Memory allocation file is printed at a width of 132 characters
XT:n	Task builder exits after n diagnostics

Defaults and the use of a minus sign (-) to negate switches are identical to RSX-11M.

Use of overlays can be avoided to take advantage of VAX/VMS virtual address space. VAX/VMS provides 65K bytes of virtual address space for RSX-11M images.

3.6.2 Task Builder Options

You can specify any of the RSX-11M task builder options for RSX-11M images that are to run under VAX/VMS. The options that specify UIC, priority, and partition have no effect; UIC and priority are associated with the user running the image in VAX/VMS, and partitions do not exist. Two options, TASK and ASG, may be required for RSX-11M images to run successfully under VAX/VMS, as described below.

RSX-11M images that use common event flags (that is, specifying event flag numbers greater than 33), that are the target of directive requests, or that issue RECEIVE DATA and RECEIVE DATA OR EXIT directives must have a name assigned using the TASK option. The first character of the name must be alphabetic. VAX/VMS uses the presence of a task name in the image label block as an indication that the image requires special initialization.

RSX-11M images that do not associate LUNs with either a physical or logical device name within the program must be built using the ASG option to make the required association. VAX/VMS MCR does not support the Reassign command.

3.6.3 Noncontiguous Image Files

The task builder issues an informational message if it produces an image file that is not contiguous. VAX/VMS, however, allows you to run images contained in noncontiguous files.

3.6.4 Building RSX-11M Images on VAX/VMS for Execution on RSX-11M

You also can use the VAX/VMS system to build RSX-11M images that are to run under RSX-11M. In this case, VAX/VMS imposes no restrictions on the switches and options that you can specify.

CHAPTER 4
MCR COMMANDS

ALLOCATE

4.1 ALLOCATE DEVICE

The Allocate Device command reserves an unmounted shareable device or an unallocated nonshareable device as a private device for the process. After the device is allocated, MCR displays the physical name of the device on SYS\$OUTPUT. Other users cannot gain access to an allocated device.

Format:

ALLOCATE device-name[:]

device-name	Specifies the physical or logical name of the device to be allocated. If you specify a physical device name, it can be either a complete name or a generic device name, for example TT. See Section 3.2.1.1 for a discussion of generic device names.
-------------	---

Examples:

```
> ALL TTH7
 _TTH7:  ALLOCATED
```

This command requests allocation of device TTH7. The Allocate command responds by indicating that the requested device has been reserved for the requesting process. The underscore preceding the displayed device name indicates that no further name translation is required to determine the actual physical device allocated.

```
> ASN MTA2:=MT0:
> ALL MT0
 _MTA2:  ALLOCATED
```

The Assign command defines MT0 as the logical name for the physical device unit MTA2. The Allocate command then reserves that device using its logical name.

Notes:

- The Assign command can be used to define a logical device name.
- Assigning an I/O channel (LUN) to a nonshareable device causes the device to be implicitly allocated. Opening a file causes a channel to be assigned.
- File-oriented devices (magnetic tape and disks) cannot be allocated if they are mounted.
- If you do not specify the device controller designation and unit number under VAX/VMS, the Allocate command selects an available device. The selected device is not necessarily controller A and unit 0.

APPEND

4.2 APPEND

The Append command adds the contents of one or more specified input files to the end of a specified output file.

Format:

```
APPEND [keywords] output-file-spec [keywords]
      = input-file-spec [keywords] ,...
```

keywords /ALLOCATION=n
 /[NO]CONTIGUOUS
 /EXTENSION=n
 /FILE_MAXIMUM=n
 /[NO]LOG
 /[NO]NEW
 /PROTECTION=code
 /[NO]READ_CHECK
 /[NO]WRITE_CHECK

output-file-spec Is the name of the output file. You must specify at least one field of the output file specification. For fields that you do not specify, the Append command uses the related field of the input file specification.

If you specify a wild card in any field of the output file specification, the Append command uses the related field of the input file specification(s).

If you specify only the device and directory portions of the output file specification, the Append command uses related fields of the input file specification.

input-file-spec Specifies the name of a file to be appended. If you specify more than one input file, separate the file specifications with either a comma (,) or a plus sign (+).

You can use a wild card in place of the file name, type, or version field. Then, all files that satisfy the remaining components are appended.

MCR COMMANDS

Command Keyword:

/LOG
/NOLOG
 (default)

Indicates that the Append command is to display the file specifications of each output file created and each input file copied in the following format.

CREATING NEW FILE
input-file COPIED TO output-file (nn RECORDS)

When the Append command is creating a single output file from multiple input files, the first message is followed by messages in the following format.

input-file APPENDED TO output-file (nn BLOCKS)

The Append command displays either the number of records or the number of blocks copied depending on whether the file is copied on a record-by-record or block-by-block basis.

At the end of command processing, the Append command displays the number of new files created, which is always one or zero.

File Keywords:

/ALLOCATION=n

Forces the initial allocation of the new output file to the number of blocks specified by n.

/ALLOCATION implies /NEW; that is, the allocation values are applied only if a new output file is actually created.

/CONTIGUOUS
/NOCONTIGUOUS

Indicates whether the output file is to be contiguous; that is, whether it must occupy consecutive physical disk blocks.

By default, the Append command creates an output file in the same format as the related input file. If multiple input files of different formats are appended to a single output file, the output file may or may not be contiguous. To ensure that the output file is contiguous, use the /ALLOCATION and /CONTIGUOUS keywords and create a new output file.

/EXTENSION=n

Specifies the number of blocks to be added to the new output file each time it is extended.

/EXTENSION implies /NEW; that is, the allocation values are applied only if a new output file is actually created.

/FILE_MAXIMUM=n

Specifies the maximum number of logical records that the output file can contain. This qualifier is valid only for relative files.

/NEW
/NONEW (default)

Requests that, if the specified output file does not already exist, the Append command create one.

MCR COMMANDS

<code>/PROTECTION=code</code>	Defines the protection to be applied to the output file. The protection is specified following the standard rules, described with the Set Protection command. Any protection attributes not specified are taken from the current protection of the output file.
<code>/READ_CHECK</code> <code>/NOREAD_CHECK</code> <code>(default)</code>	Indicates whether the Append command is to read the specified input file(s) twice to verify that all records have been correctly read.
<code>/WRITE_CHECK</code> <code>/NOWRITE_CHECK</code> <code>(default)</code>	Indicates whether the Append command is to read the output file after it has been written to verify that the file has been successfully appended and that the output file can be read without error.

Examples:

```
> APPEND [HICKS]EDWARD.OBJ = [DAVIS]JOSEPH.OBJ
```

This command appends the file JOSEPH.OBJ from directory [DAVIS] to the file EDWARD.OBJ in directory [HICKS] on the default device.

```
> APPEND /LOG/NEW THEEIGHT.DAT = HENRI.DAT+LUKS.DAT+SLOAN.DAT
%APPEND-I-CREATED, DBB1:[PAINT]THEEIGHT.DAT;1 created
%APPEND-S-COPIED, DBB1:[PAINT]HENRI.DAT;2 copied to DBB1:[PAINT]
THEEIGHT.DAT;1 (10 records)
%APPEND-S-APPENDED, DBB1:[PAINT]LUKS.DAT;5 appended to DBB1:[PAINT]
THEEIGHT.DAT;1 (8 records)
%APPEND-S-APPENDED, DBB1:[PAINT]SLOAN.DAT;12 appended to
DBB1:[PAINT]THEEIGHT.DAT;1 (22 records)
%APPEND-S-NEWFILES, 1 file created
```

This command concatenates the files HENRI.DAT, LUKS.DAT, and SLOAN.DAT into a new file named THEEIGHT.DAT. The input and output files are on the default device and directory.

Notes:

- RSX-11M does not support the Append command.
- The Append command is a variation of the Copy command.
- When you use wild cards to concatenate Files-11 Structure Level 1 files, the input files are copied to the output file in random order. When you use wild cards to concatenate Files-11 Structure Level 2 files, the Append command copies files in alphanumeric order. When wild cards are used in the version field, files are copied in descending order by version number; that is, files with the same name are copied starting with the highest version number.
- Additional examples of appending files are provided in the VAX/VMS Command Language User's Guide.

ASN

4.3 ASSIGN

The Assign command defines or deletes a logical name assignment for a device in one of three logical name tables: process, group, or system.

Format for Assigning a Logical Name:

ASN [keywords] ppnn:=llnn:

Format for Deleting a Logical Name:

ASN =llnn: [keywords]

keywords /GBL
 /GROUP
 /TERM

pp Indicates the equivalence (physical, logical, or pseudo-device) name to which the logical name is to be assigned. The equivalence name can be up to 15 characters.

nn Specifies the device unit number.

ll Specifies a logical name of up to 15 characters.

Command Keywords:

/GBL Indicates that the requested action is to occur in the system logical name table.

/GROUP Indicates that the requested action is to occur in the group logical name table.

/TERM Indicates that the requested action is to occur in the process logical name table. /TERM is the default.

Examples:

> ASN MTB2:=MT0:

This command assigns the string MT0 as the logical name for device MTB2 in the process logical name table.

> SHOW LOGICAL MT0
MT0 = MTB2: (PROCESS)

This command requests that the equivalence name associated with logical name MT0 be displayed.

MCR COMMANDS

Notes:

- The equal sign (=) between the two name parameters is optional; if you do not specify the equal sign, you must leave a space between the two parameters.
- RSX-11M does not support the /GROUP keyword.
- VAX/VMS does not support the /LOGIN and /TERM:ttn keywords.
- Privilege is required to assign and delete logical names in the group or system logical name tables.
- The Show Logical command can be used to display logical names in the three tables.

BYE

4.4 BYE

The Bye command terminates an interactive terminal session. The system displays a termination message and performs any necessary cleanup operations, such as terminating the current image if one exists, dismounting any private volumes that remain mounted, and deallocating devices. Finally, it deletes your process and subprocesses, if any.

Format:

BYE

Example:

```
> BYE
MURPHY      logged out at 4-APR-1978 13:33:26.57
```

Note:

- The Logout command performs the same function as the Bye command.

CANCEL

4.5 CANCEL

The Cancel command requests the system to cancel a wake request for a specified process.

A wake request can be the result of any of the following:

- The RUN\$ directive (see Notes below)
- The Schedule Wakeup system service
- Any of the time-synchronized forms of either the MCR or DCL RUN commands

Cancel does not affect the execution of an active process; it only removes time-based requests from the timer queue.

Format:

CANCEL [keyword] [process-name]

keyword /IDENTIFICATION=pid

process-name Specifies the name of the process for which wake requests are to be canceled. Because process names are implicitly qualified by UIC group number, you can cancel requests only for process within your group by specifying a process name.

Command Keyword:

/IDENTIFICATION=pid Specifies the process identification of the process for which wake requests are to be canceled. If you specify both a process name and identification, the Cancel command ignores the process name. If you specify neither, wake requests for your current process are canceled.

Example:

> CAN PROCA

This command cancels wake requests for a process executing in the same group as the requester. If the process is not one of the requester's subprocesses, the requester must have group privilege.

> CAN /IDENTIFICATION=C0025

This command cancels wake requests for the process whose identification is C0025. It can be executing outside of the requester's group; appropriate privilege is required, however.

MCR COMMANDS

Notes:

- You must have group or world privilege to cancel wake requests for another process. No privilege is required for a process to issue a cancel request for one of its subprocesses.
- The Stop command can be used to delete a hibernating process for which all wakeup requests are canceled.
- You can determine whether a subprocess has completed execution of an image by issuing the Show Process command with the /SUBPROCESSES keyword.
- VAX/VMS uses the Schedule Wakeup system service to emulate the RSX-11M RUN\$ directive; for additional information, refer to the VAX-11/RSX-11M Programmer's Reference Manual.

4.6 CONTINUE

CONTINUE

The Continue command resumes execution of a command, program, or indirect command file that was interrupted by any of the following:

- Pressing CTRL/Y or CTRL/C
- A PAUSE request issued by a FORTRAN program
- A .PAUSE directive issued in an indirect command file

When a program is interrupted or pauses, only those actions listed below can be taken and still be able to resume execution; unlisted commands cause termination of the interrupted image:

- Examine and Deposit
- Allocate and Deallocate
- Define a symbol (.SETx)
- Assign and delete a logical name
- Set and show the default directory and default file protection
- Show the translation of a logical name
- Display the time

When an indirect command file pauses, VAX/VMS places no restrictions on the commands that you can use. Restrictions on the use of the .PAUSE directive are presented in Section 5.11.

Format:

CONTINUE

Example:

```
> RUN TEST
.
.
.
^Y
> TIME
4-APR-1978 14:00:55
> CONTINUE
```

The Run command initiates execution of the image TEST. Pressing CTRL/Y interrupts TEST. The Time command causes VAX/VMS to display the date and time. The Continue command then requests resumption of TEST at the point where it was interrupted.

Notes:

- RSX-11M does not support this command.
- For convenience, the Continue command can be abbreviated to the single character, C.

COPY

4.7 COPY

The Copy command creates a new file from one or more existing files. The Copy command can be used to perform the following functions.

- Copy one file to another file
- Concatenate more than one file into a single output file
- Copy a group of files to another group of files

Format:

```
COPY [keywords] output-file-spec [keywords]
      = input-file-spec [keywords],...
```

keywords /ALLOCATION=n
 /[NO] CONCATENATE
 /[NO] CONTIGUOUS
 /EXTENSION=n
 /FILE MAXIMUM=n
 /[NO] LOG
 /[NO] OVERLAY
 /PROTECTION=code
 /[NO] READ_CHECK
 /[NO] REPLACE
 /[NO] TRUNCATE
 /[NO] WRITE_CHECK

output-file-spec Is the name of the output file. You must specify at least one field of the output file specification. For fields that you do not specify, the Copy command uses the related field of the input file specification(s).

If you specify a wild card in any fields of the output file specification, the Copy command uses the related field of the input file specification(s) and does not concatenate the output.

If you specify only the device and directory portion of the output file specification, the Copy command uses related fields of the input file specification(s) and does not concatenate the output.

input-file-spec Specifies the name of a file to be copied. If you specify more than one input file, separate the file specifications with either a comma (,) or a plus sign (+).

You can use a wild card in place of the file name, type, or version field. Then, all files that satisfy the remaining components are copied.

MCR COMMANDS

Command Keywords:

/LOG
/NOLOG
 (default)

Indicates that the Copy command is to display the file specifications of each input file copied in the following format.

input-file COPIED TO output-file (nn RECORDS)

When the Copy command is creating a single output file from multiple input files, the first message is followed by messages in the following format.

input-file APPENDED TO output-file (nn BLOCKS)

The Copy command displays either the number of records or the number of blocks copied depending on whether the file is copied on a record-by-record or block-by-block basis.

At the end of command processing, the Copy command displays the number of files created.

/CONCATENATE
 (default)
/NOCONCATENATE

When multiple input file specifications are provided, this keyword indicates whether a single output file is to be created from all files that satisfy the input file specification.

By default, the Copy command produces a single output file from multiple input files unless:

- One or more wild cards are present in the output file specification.
- Only the device and directory portions of the output file specification are provided.

File Keywords:

/ALLOCATION=n

Forces the initial allocation of the output file to the number of blocks specified by n. If this keyword is not specified, the initial allocation of the output file is determined from the size of the input file being copied.

/CONTIGUOUS
/NOCONTIGUOUS

Indicates whether the output file is to be contiguous; that is, whether it must occupy consecutive physical disk blocks.

By default, the Copy command creates an output file in the same format as the corresponding input file. If the input file is contiguous, the Copy command attempts to create a contiguous output file, but does not report an error if the output file is not contiguous.

If multiple input files of different formats are copied into a single output file, the output file may or may not be contiguous. To ensure that the output is contiguous, use the /ALLOCATION and /CONTIGUOUS keywords.

MCR COMMANDS

/CONTIGUOUS /NOCONTIGUOUS (Cont.)	/CONTIGUOUS has no effect when you copy files to or from magnetic tapes because the size of the input file cannot be determined. If you copy a file from tape, use two Copy commands: one to copy the file from tape, and another to create a contiguous file.
/EXTENSION=n	Specifies the number of blocks to be added to the output file each time it is extended. If this keyword is not specified, the default extension attribute of the output file is determined from the input file.
/FILE_MAXIMUM=n	Specifies the maximum number of logical records that the output file can contain. This qualifier is valid only for relative files.
/OVERLAY /NOOVERLAY (default)	Indicates whether data in the input file is to be copied into an existing output file overlaying the existing data. If the new file is shorter than the overlaid file, the file is truncated at the end of the new file. The new file is in the same position as the overlaid file. /OVERLAY is ignored if the output file is written to a nonfile-structured device.
/PROTECTION=code	Defines the protection to be applied to the output file. The protection is specified following the standard rules, described with the Set Protection command. Any protection attributes not specified are taken from the current protection of the input file.
/READ_CHECK /NOREAD_CHECK (default)	Indicates whether the Copy command is to read the specified input file(s) twice to verify that all records have been correctly read.
/REPLACE /NOREPLACE (default)	Requests that, if a file already exists with the same file specification as that entered for the output file, the existing file be deleted and replaced with the contents of the input file(s). By default, the Copy command creates a new version of a file if the file already exists.
/TRUNCATE /NOTRUNCATE (default)	Indicates whether the command is to copy all blocks allocated to the file or only those blocks that contain data. When copying multiple input files into one output file, you can save space by specifying /TRUNCATE. If you do not specify /TRUNCATE, all of the blocks allocated to the specified file(s) are copied to the output file(s).
WRITE_CHECK /NOWRITE_CHECK (default)	Indicates whether the Copy command is to read the output file after it has been written to verify that the file has been successfully copied and that the output file can be read without error.

MCR COMMANDS

Examples:

> COPY [HICKS]EDWARD.OBJ = [DAVIS]JOSEPH.OBJ

This command copies the file JOSEPH.OBJ from directory [DAVIS] to directory [HICKS] on the default device and names the file EDWARD.OBJ.

> COPY ALLSRC.FOR = *.FOR

This command concatenates the highest versions of all FORTRAN source files in the default directory into a file named ALLSRC.FOR.

Notes:

- RSX-11M does not support the Copy command; however, under VAX/VMS, you can use PIP instead of Copy.
- The Append command is a variation of the Copy command.
- When you use wild cards or /CONCATENATE to concatenate Files-11 Structure Level 1 files, the input files are copied to the output file in random order. When you concatenate Files-11 Structure Level 2 files, the Copy command copies files in alphanumeric order. When wild cards are used in the version field, files are copied in descending order by version number; that is, files with the same name are copied starting with the highest version number.
- The VAX/VMS Command Language User's Guide provides additional examples of the use of the Copy command.

DEALLOCATE

4.8 DEALLOCATE

The Deallocate command releases a previously allocated device. The system automatically deallocates any allocated devices when the owner logs off.

Format:

```
DEALLOCATE [keyword] [device-name]
```

keyword /ALL

device-name Specifies the physical or logical name assigned of the device to be deallocated. If you omit the controller designator and unit number is a physical device name, they default to controller A and unit 0, respectively.

Command Keyword:

/ALL Requests that all devices currently allocated to your process be deallocated. If you specify /ALL, you cannot specify a device name.

Example:

```
> DEA MT0
```

This command deallocates the physical device whose name is the equivalence for the logical name MT0.

DEBUG**4.9 DEBUG**

The Debug command invokes a debugger after you have interrupted a program's execution by pressing CTRL/C or CTRL/Y. If the image is native, the VAX-11 Symbolic Debugger prompts. If it is an RSX-11M image that was built with ODT, ODT prints a T-bit exception message.

Format:

DEBUG

Example:

```
> RUN COPLEY/NODEBUG
^Y
> DEBUG
DBG>
```

In the example above, typing CTRL/Y interrupts execution of a native image and causes MCR to prompt. Typing the Debug command at this point causes the VAX-11 debugger to prompt.

```
> RUN PEALE/NODEBUG
^Y
> DEBUG
TE:011454
```

In the example above, typing CTRL/Y interrupts execution of an RSX-11M image and causes MCR to prompt. Typing the Debug command at this point causes ODT to prompt.

Notes:

- If no image is currently executing, the Debug command performs no operation.
- If the interrupted program was not linked with the appropriate debugging aid (VAX-11 Symbolic Debugger for native images or ODT for RSX-11M images), the Debug command causes a software exception condition.

If the image is a native image that has not declared a condition handler, or if it is an RSX-11M image that has not specified a T-bit SST routine address, this exception may cause termination of the image.

- For details on the use of the debugger, refer to the VAX-11 Symbolic Debugger Reference Manual.
- For details on condition handling and exception conditions, refer to the VAX/VMS System Services Reference Manual.

DELETE

4.10 DELETE

The Delete command deletes files, entries from a printer queue, and entries from a batch job queue, and stops the printing of a file.

You specify /ENTRY to delete a batch or printer queue entry. You can specify any of the remaining keywords to control the deletion of files. The /CREATED, /MODIFIED, and /EXPIRED keywords determine whether files are to be deleted based on their creation date, most recently modified date, or planned expiration date. The /BEFORE and /SINCE keywords are used in conjunction with the selection date keywords to establish the time span to be considered by the Delete command.

Format:

DELETE [keywords] file-spec,...

keywords	/BEFORE [= (abs-time)] /[NO]CONFIRM /CREATED /ENTRY=jobid,... /EXPIRED /[NO]LOG /MODIFIED /SINCE [= (abs-time)]
----------	--

file-spec	Specifies the name of a file to be deleted. If you specify more than one file specification, separate them with a comma (,) or a plus sign (+).
-----------	---

The first file specification must contain a file name, file type, and version number. You can specify these fields as wild cards. Subsequent file specifications can omit any of these fields except version number; the first file specification provides defaults for additional file specifications.

If you specify /ENTRY, the file specification is the name of the queue in which a printer or batch job was entered.

Command Keywords:

/BEFORE [= (abs-time)] /BEFORE [=TODAY]	Chooses only those specified files that have a selection date before the specified date. The selection date can be specified either as an absolute date or as TODAY. Section 2.7.6.1 explains how to specify an absolute time.
--	--

If you do not specify /BEFORE, the command deletes all specified files regardless of date, unless /SINCE is specified.

If you specify /BEFORE but do not supply a selection date, the Delete command uses TODAY by default; that is, the command deletes all files created or modified yesterday or before.

MCR COMMANDS

`/CONFIRM`
`/NOCONFIRM (default)` Requests the command to display the name of each file before it is deleted and wait for you to type a confirmation. If you type Y in response to the message, the file is deleted. Any other response causes the file to be retained.

`/CREATED` Requests that specified file be selected for deletion based on their creation date. This keyword is meaningful only when `/BEFORE` or `/SINCE` is specified.

The default selection date is both the created date and the most recently modified date.

`/ENTRY=jobid,...` Specifies the job identification of the job to be deleted from a printer or batch job queue. If you specify more than one job identification, separate them with a comma (,) and enclose the list in parentheses.

The system assigns a unique entry number, called a job identification, to each spool printer or batch job in the system. The job identification is displayed by the Print and Submit commands when the job is queued. You also can determine the job identification of a queued job using the Show Queue command.

If you specify `/ENTRY`, no other keywords can be specified.

`/EXPIRED` Selects specified files according to their planned expiration date. This keyword is meaningful only when `/BEFORE` or `/SINCE` is specified.

The default selection date is both the created date and the most recently modified date.

`/LOG`
`/NOLOG (default)` Determines whether the file specification of each file is to be displayed as the file is deleted. If you specify `/LOG`, the delete command displays the following information:

`devcu:[dir]filename.type;version deleted`

`/MODIFIED` Selects specified files according to the date on which they were most recently modified. This keyword is the default and is meaningful only when `/BEFORE` or `/SINCE` is specified.

`/SINCE[=(abs-time)]`
`/SINCE[=YESTERDAY]` Chooses only those specified files that have a selection date after the specified date. The selection date can be specified either as an absolute time or as YESTERDAY. Section 2.7.6.1 explains how to specify an absolute time.

If you do not specify `/SINCE`, the command deletes the specified files regardless of date, unless `/BEFORE` is specified.

If you specify `/SINCE` but do not supply a selection date, the Delete command uses YESTERDAY by default; that is, the command deletes all specified files with a selection date of today.

MCR COMMANDS

Examples:

> DELETE /SINCE=(11:30) INPUT.DAT;*

This command deletes all versions of the file INPUT.DAT created or modified after 11:30 today.

> DELETE/ENTRY=243 SYS\$PRINT

This command deletes from the printer queue named SYS\$PRINT the print job with the identification 243. If the job is currently being printed, printing stops.

> DELETE/ENTRY=634 SYS\$BATCH

This command deletes from the batch queue named SYS\$BATCH the batch job with the identification 634.

Note:

- RSX-11M does not support this command.

DEPOSIT**4.11 DEPOSIT**

The Deposit command replaces the contents of a specified location in virtual memory. After the contents are replaced, Deposit displays the location and the newly deposited data as follows:

address: contents

The Deposit command converts the specified value to binary before placing it in virtual memory.

If the address specified can be read, but not written, the current contents of the location are not altered, but are displayed. If the address specified cannot be read or written, asterisks (****) are displayed.

When used with the Examine command, the Deposit command can aid in the debugging of programs interactively. It is not necessary to link with a debugger to use these commands.

Format:

DEPOSIT [keywords] address=data [,...]

keywords /ASCII
 /BYTE
 /DECIMAL
 /HEXADECIMAL
 /LONGWORD
 /OCTAL
 /WORD

address The virtual address of a location whose contents are to be changed. The specified location must be within the virtual address space of the image currently running in the process, and it must be read/write for user access mode.

You can specify the location as a symbolic name that has been assigned a value equivalent to a virtual memory location. The arithmetic operators +, -, *, and / can be used to express the location. The value of a symbol used in an expression always is interpreted in the radix used to define it. Any numeric value used in the expression is interpreted according to the current default radix of the command. Refer to the examples below.

The Deposit and Examine commands set a pointer to the byte following the last byte modified. You can refer to this pointer as "." for the first location in a subsequent Examine command or as the deposit location in a subsequent Deposit command.

MCR COMMANDS

data Data to be deposited into the specified location. If you specify a list of data, the data is deposited into consecutive locations beginning at the specified location. By default, data is assumed to be in hexadecimal format.

The Deposit command converts the specified data to binary and writes it into the specified virtual memory location.

Command Keywords:

/ASCII Indicates that the data specified is in ASCII format.

When ASCII is specified or is the default, hexadecimal is the default radix for numeric literals. The address specified is interpreted as a hexadecimal address.

/BYTE Specifies that data is to be deposited one byte at a time. If the data specified is longer than a byte, an error message is displayed.

/DECIMAL Indicates that the default radix for numeric literals and displayed values is decimal.

/HEXADECIMAL Indicates that the default radix for numeric literals and displayed values is hexadecimal. This is the initial default of the command; refer to notes below.

/LONGWORD Specifies that data is to be deposited one longword at a time. This is the initial default of the command; refer to notes below.

/OCTAL Indicates that the default radix for numeric literals and displayed values is octal.

/WORD Specifies that data is to be deposited one word at a time.

Examples:

```
> RUN    RSXPROG
      .
      .
      .
      ^Y
> EXAMINE/OCT/WORD 2332
000004DA:    007402
> DEPOSIT . = 777
000004DA:    000777
> CONTINUE
```

CTRL/Y interrupts execution of the RSX-11M image RSXPROG. The Examine command establishes the radix as octal and the length unit as a word and displays the content of location 4DA (hexadecimal). The Deposit command then deposits the octal value 777 in location 4DA.

MCR COMMANDS

```
> .SETN BASE %X200
> RUN XYZ
.
.
^Y
> EXAMINE BASE:BASE+20
00000200: 00037C02 0003CFF0 6D756E20 2C726562 646E6520 74697720 39392068
0000021C: 0F13CBA1 0C07040A
> DEPOSIT BASE+16=0723C1D0
> CONTINUE
```

In the sequence above, the .SETN directive, typed interactively at the terminal, defines the symbol BASE as being equal to 200 (hexadecimal). Refer to Section 5.7 for a description of numeric literals under VAX/VMS. The Examine command displays the contents of locations 200 (hexadecimal) through 220 (hexadecimal). The Deposit command deposits data in location 216 (hexadecimal).

In the example above, hexadecimal is the default radix; therefore, data is deposited at location 216 (hexadecimal). If the default were octal, the data would be deposited at the location derived from 200 (hexadecimal) plus 16 (octal).

Notes:

- The initial default radix for the Deposit command is hexadecimal. The default applies to both the interpretation of numeric literals in the command line and to the data. If you use a radix keyword to modify the command, that radix becomes the default for subsequent Deposit and Examine commands.
- The initial default length unit for the Deposit command is a longword. The Examine command displays data one longword at a time with blanks between longwords. If you use a /BYTE or /WORD keyword to modify the command, that unit becomes the default for subsequent Deposit and Examine commands.
- RSX-11M images start at virtual address 0 under VAX/VMS; this is the same starting address that they have under RSX-11M.
- RSX-11M does not support this command.
- For convenience, the Deposit command can be abbreviated to the single character, D.

DIRECTORY

4.12 DIRECTORY

The Directory command provides a list of files or information about a file or group of files.

Format:

DIRECTORY [keywords] [file-spec,...]

keywords /BRIEF
 /FULL
 /OUTPUT=file-spec
 /PRINTER

file-spec Specifies the names of files to be listed. The syntax of a file specification determines what file(s) are listed, as follows:

- If you do not enter a file specification, the Directory command lists all files in your current default directory.
- If you specify only a device name, the Directory command uses your default directory name.
- Whenever the file specification does not include a file name and file type, all versions of all files in the specified directory are listed.
- If a file specification contains a file name and/or file type and no version number, the Directory command lists the highest existing version.
- If a file specification contains only a file name, the Directory command assumes a null file type; an unpunctuated item is assumed to be a file name.

If you specify more than one file, separate the file specifications with either a comma (,) or plus sign (+). You can use wild cards in place of the directory, file name, file type, or version fields of a file specification to list all files that satisfy the components you specify.

Command Keywords:

/BRIEF Lists only the file name, type, and version of each file specified.

/FULL Lists the following information about the specified files:

- Disk and directory name
- Date and time that the Directory command was

MCR COMMANDS

- File name, type, and decimal version number of each file
- File identification number in the format:
(file-number,file-sequence-number)
- Number of blocks occupied by the file
- Number of blocks allocated for the file
- file code, as follows:
null -- file is noncontiguous
C -- file is contiguous
L -- file is locked
- Date and time that the file was created
- UIC of the file's owner and the protection code associated with the file in the format:
[group,member] [system,owner,group,world]
- Summary information in the format:
TOTAL OF in-use./allocated BLOCKS IN xxxxx.
FILES

/OUTPUT=file-spec Requests that the Directory command output be written to the file specified rather than to the current SYS\$OUTPUT device.

/PRINTER Queues a copy of the command output to the system printer. The printer file has a file name of DIRECT and a type of LOG.

Examples:

> DIRECTORY *.CMD;*

```
DIRECTORY DBA1:[122,20]
08-JUL-78 05:30
```

```
LOGIN.CMD;3      1.    29-FEB-76   15:36
COMPILE.CMD;15   4.    04-JAN-78   07:10
COMPILE.CMD;16   4.    04-JAN-78   07:15
```

This command displays the default directory information for all versions of all files with a file type of CMD in the default directory.

> DIRECTORY/FULL *.CMD

```
DIRECTORY DBA1 :[122,20]
08-JUL-78 11:40
```

```
LOGIN.CMD;3      (556,22)  1./1.    29-FEB-76 15:36 [122,20] [RWED,RWED,R,R]
COMPILE.CMD;16   (6453,13) 4./5.    04-JAN-78 07:15 [122,20] [RWED,RWED,RWED,R]
```

This command displays full directory information for the highest version of all files with a file type of CMD in the default directory.

MCR COMMANDS

Notes:

- You can use the /LI switch with PIP to perform the same functions as the Directory command.
- RSX-11M does not support the Directory command.

DMOUNT**4.13 DISMOUNT**

The Dismount command releases volumes previously specified in a Mount command.

Format:

DMOUNT[keyword] device-name[:]

keyword /[NO] UNLOAD

device-name Specifies the name of the device to be dismounted. The device name can be a physical device name or a logical name.

If you specify a colon following the device name, the command interpreter strips the colon.

Command Keyword:

/UNLOAD (default) Indicates that the device is to be physically unloaded. /UNLOAD is equivalent to pressing the button on the drive to unload the volume. /UNLOAD is the default action; specify /NOUNLOAD if you want the unit to remain loaded.

Example:

```
> MOU/SHARE DBA3:PROJ123 DISK
.
.
.
DMO DISK
```

The Mount command requests access to the volume labeled PROJ123 on DBA3 and places the logical name DISK in the current process's logical name table. Because the volume was requested using the /SHARE keyword, the volume may or may not already be mounted by other users.

The Dismount command releases access to the volume PROJ123 for the current process and removes the name DISK from the process logical name table. The actual dismounting does not occur until all accessors have issued dismount commands.

Notes:

- If the volume has been mounted /SHARE, it is not actually dismounted until all users who have mounted it have dismounted it.
- Dismounting a volume causes the logical name assigned to the volume by the previous Mount command to be removed from the appropriate logical name table. The logical name is either the one specified in the Mount command or the default name of DISK\$volume-label or TAPE\$volume-label.

MCR COMMANDS

- If a volume is mounted /SYSTEM or /GROUP, a dismount command causes the volume to be dismounted even if others are currently using it. However, dismounting the volume also results in removing the name from the system or group logical name table, which requires privilege.
- If the device has been allocated with an Allocate command, it remains allocated after the volume is dismounted.
- If the volume has been mounted using the /NOSHARE keyword, the system automatically allocates the device at mount time and deallocates it when the volume is dismounted.
- RSX-11M supports the Dismount command.

EXAMINE

4.14 EXAMINE

The Examine command displays the contents of virtual memory at the terminal. It is used after an RSX-11M or native image has been interrupted by CTRL/Y. If you interrupt an image while it is executing, you can examine its locations; then you can resume image execution by issuing a Continue command.

The information is displayed in the following format.

location: content

Information that is inaccessible at user mode is displayed as asterisks (****).

When used with the Deposit command, the Examine command can aid in the debugging of programs interactively. It is not necessary to link the image with a debugger to use these commands.

Format:

EXAMINE [keywords] location[:location]

keywords /ASCII
 /BYTE
 /DECIMAL
 /HEXADECIMAL
 /LONGWORD
 /OCTAL
 /WORD

location Specifies the address or range of addresses in virtual memory to be displayed. If you specify a range of addresses, you must separate the two addresses with a colon (:). Locations can be specified using expressions that contain the arithmetic operators +, -, *, and /. A symbolic name also can be used to express an address. The value of a symbol used in an expression always is interpreted in the radix used to define it. Any numeric value used in the expression is interpreted according to the current default radix of the command. Refer to the examples below.

Addresses always are displayed in hexadecimal radix.

The Examine and Deposit commands set a pointer to the last byte examined; you can refer to this pointer as "." when specifying the first location in a subsequent Examine command or as the deposit location in a subsequent Deposit command.

MCR COMMANDS

Command Keywords:

/ASCII	Indicates that data at the specified location is to be displayed in ASCII representation. Binary values that do not have ASCII equivalents are displayed as periods (.). When ASCII is specified or is the default, hexadecimal is the default radix for numeric literals.
/BYTE	Indicates that data at the specified location is to be displayed one byte at a time.
/DECIMAL	Indicates that the data is to be displayed in decimal representation and changes the default radix for numeric literals.
/HEXADECIMAL	Indicates that the data is to be displayed in hexadecimal representation and changes the default radix for numeric literals. This is the initial default of the command; refer to notes below.
/LONGWORD	Indicates that data at the specified location is to be displayed one longword at a time. This is the initial default of the command; refer to notes below.
/OCTAL	Indicates that the data is to be displayed in octal representation and changes the default radix for numeric literals.
/WORD	Indicates that data at the specified location is to be displayed one word at a time.

Examples:

```
> RUN RSXPROG
.
.
.
^Y
> EXAMINE/WORD/OCT 2677
000005BF: 007402
> CONTINUE
```

The RUN command begins execution of the RSX-11M image RSXPROG. While the image is running, the CTRL/Y function interrupts it. The Examine command requests that the contents of the word addresses by virtual memory location 5BF (hexadecimal) be displayed in octal. After the display, the Continue command causes image execution to be resumed.

MCR COMMANDS

```
> RUN VAXPROG
.
.
.
^Y
> EXAMINE/LONG/ASC 1C00+50:1C00+7F
00001C50:  FOOLISH CONSISTENCY IS THE HOBGOBLIN OF LITTLE M
> EXA 1C80:1C83
00001C80:  INDS
```

While the native image VAXPROG is executing, it is interrupted by CTRL/C. The Examine command then requests that all the data in the virtual address range specified be displayed in ASCII representation. Note that the addresses are expressed in hexadecimal and are interpreted as such.

```
> RUN ABC
.
.
.
^Y
> .SETN BASE %X200
> EXAMINE BASE:BASE+20
00000200: 143F4D00 1537FF3C ...
0000021C: 00032479
> CONTINUE
.
.
.
^Y
> EXAMINE/WORD/OCTAL BASE:BASE+20
00000200: 004672 467013 ...
> CONTINUE
```

In the sequence above, the .SETN directive, typed interactively at a terminal, defines the symbol BASE as being equal to 200 (hexadecimal). The first Examine command displays the contents of location 200 (hexadecimal) through 220 (hexadecimal). The contents of 32 locations are displayed.

In the second Examine command, the content of 16 locations are displayed. The Examine command uses the current radix (octal) as the radix for the numeric value (20) in the expression.

Notes:

- The initial default radix for the Examine command is hexadecimal. The default applies to both the interpretation of numeric literals in the command line and to the representation of the data displayed. If you use a radix keyword to modify the command, that radix becomes the default for subsequent Examine and Deposit commands.
- The initial default length unit for the Examine command is a longword. The Examine command displays data one longword at a time with blanks between longwords. If you use a /BYTE or /WORD keyword to modify the command, that unit becomes the default for subsequent Examine or Deposit commands.
- RSX-11M images start at virtual address 0 under VAX/VMS; this is the same starting address that they have under RSX-11M.
- For convenience, the Examine command can be abbreviated to a single character, E.

Execute Procedure (@)

4.15 EXECUTE PROCEDURE (@)

The Execute procedure (@) requests execution of an indirect command file or requests the command interpreter to read subsequent command input from a specified file or device.

Format:

```
@file-spec [keyword] [p1 [p2 [...p8]]]
```

keyword /OUTPUT=file-spec
 /DELETE

file-spec Specifies the command procedure to be executed, or the device from which input for the preceding command is to be read.

If you do not specify a file type, the MCR command interpreter uses the default file type CMD.

p1,p2,...p8 Specify from one to eight optional parameters to pass to the indirect command file. The parameters assign numeric or character string values to the symbols, P1, P2, and so on up to P8 in the order of entry. The symbols are local to the indirect command file. Unspecified parameters are set to null strings. Separate each parameter with one or more blanks.

You can specify a numeric value for a parameter using any valid arithmetic expression. You also can specify a character string value using any alphanumeric or special characters, with the following restrictions:

- If the first parameter begins with a slash (/), you must enclose the parameter in quotation marks.
- To pass a parameter that contains embedded blanks, place the parameter in quotation marks.
- To pass a parameter that contains literal quotation marks, enclose the entire string in quotation marks and use a double set of quotation marks within the string, for example:

```
  "NEVER SAY ""DIE"""
```

In each case above, the command interpreter strips the set of quotation marks that enclose the entire string.

MCR COMMANDS

Command Keyword:

/OUTPUT=file-spec Requests that all output directed to the logical device SYS\$OUTPUT be written to the file or device specified. System responses and error messages are written to the terminal as well as to the specified file.

If you specify /OUTPUT, the keyword must follow the file specification of the indirect command file.

/DELETE Requests that the indirect command file be deleted after it is closed.

Examples:

> @STDJOB.CMD PAYROL FICA

This command requests execution of the indirect command file PAYROL.CMD and provides two parameters, PAYROL and FICA. The indirect command file contains the following commands.

```
RUN 'P1
PRINT 'P1'.DAT
RUN 'P2
PRINT 'P2'.DAT
```

When the indirect command file is executed, the two parameters are substituted into the Run and Print commands. The result is that the indirect command file runs PAYROL.EXE and FICA.EXE and prints the files PAYROL.DAT and FICA.DAT.

Notes:

- You can cause symbol substitution by enclosing a symbol in single quotation marks. Refer to the VAX/VMS Command Language User's Guide for further information on symbols and substitution.
- When an indirect command file terminates, any data files that remain open are closed by the MCR command interpreter.

EXIT

4.16 EXIT

The Exit command terminates the processing of the current indirect command file. If the indirect file was executing within another indirect command file, control returns to the outer file.

The Exit command is used primarily to terminate execution of the current indirect file and also maintain the status as it was prior to execution of the Exit command.

Format:

```
EXIT      [status-code]
```

status-code Defines a value for the symbol \$STATUS, which is used as a return code to be tested by the next higher command level.

If you do not specify a status code, the current value of the symbol \$STATUS is not changed and control returns to the outer level with the status of the most recently executed command or program.

The command interpreter uses numeric values returned by commands or programs to locate and display error messages. Each system message has a unique value associated with it. If you specify an even numeric value as a status code, it may cause the command interpreter to display a message that you would not expect.

If you specify an odd value for a status code, the command interpreter does not display a message.

Example:

```
>@LEVEL 1
```

This command requests execution of the indirect command file LEVEL1.CMD. It, in turn, contains a command requesting execution of the indirect command file LEVEL2.CMD. LEVEL2.CMD contains the following sequence, which causes a return to LEVEL1.CMD.

```
ON WARNING .GOTO ABEND
.
.
.
RUN LAST
EXIT
.ABEND: EXIT 7
```

The LEVEL2 command file contains an ON command that causes a branch to label ABEND if any error occurs. An error results in an exit from LEVEL2 with \$STATUS having a value of 7. If no error occurs, LEVEL2 exits with the status value supplied by LAST.

MCR COMMANDS

When LEVEL1 regains control, it tests the value of \$STATUS using the following command.

```
.IF $STATUS EQ 7 EXIT.
```

If \$STATUS is equal to 7, that is, if an error occurred in the LEVEL2 command file, LEVEL1 exits; otherwise, execution of LEVEL2 commands continues.

Notes:

- Specifying the Exit command is equivalent using a slash (/) in an RSX-11M indirect file. VAX/VMS also supports the use of slashes in indirect files.
- RSX-11M does not support the Exit command.

INITIALIZE

4.17 INITIALIZE

The Initialize command formats and writes a label on a mass storage volume. The default format for disk volumes created using the MCR command Initialize is Files-11 Structure Level 1. The Initialize command also can initialize Files-11 Structure Level 2 volumes. The default format for magnetic tape volumes is the ANSI standard for tape labels, Level III.

You do not need any special privileges to initialize a blank disk or tape volume. If a volume has previously been written, however, your UIC must match the owner UIC on the volume, or you must have the user privilege to override volume protection. In the case of a tape that has been written, you can initialize the volume if you are allowed write access to it.

Format:

```
INITIALIZE [keywords] device-name: volume-label
```

```
keywords      /ACCESSED=n (D)
               /BADBLOCKS=list (D)
               /CLUSTER SIZE=n (D)
               /DATA CHECK[=options,...] (D)
               /DENSITY=n (T)
               /DIRECTORIES=n (D)
               /EXTENSION=n (D)
               /FILE PROTECTION=code (D)
               /GROUP (D)
               /HEADERS=n (D)
               /INDEX=position (D)
               /MAXIMUM FILES=n (D)
               /OVERRIDE=options,... (T)
               /OWNER UIC=uic
               /PROTECTION=code
               /SHARE (D)
               /STRUCTURE=level (D)
               /SYSTEM (D)
               /USER NAME=string (D)
               /[NO]VERIFIED (D)
               /WINDOWS=n (D)
```

D = applicable to disk only.

T = applicable to tape only.

Keyword descriptions are categorized accordingly below.

device-name Specifies the name of the device on which the volume to be initialized is physically mounted. The device does not have to be allocated first; however, it is the recommended practice.

volume-label Specifies the label to be written on the volume. For a disk volume, you can specify a maximum of 12 alphanumeric characters; for a tape volume, you can specify a maximum of 6 alphanumeric characters.

MCR COMMANDS

Keywords Applicable to Disk and Tape:

/OWNER_UIC=[g,m] Specifies the user identification code to be assigned ownership of the volume and of system files on the volume. The group (g) and member (m) fields of the UIC can have a value in the range 0 through 377 (octal).

The brackets are required.

If you do not specify **/OWNER_UIC**, your current UIC is assigned ownership of the volume.

/PROTECTION=code Specifies the protection to be applied to the volume. The protection determines which users can read files, write files, create directories, and delete files on the volume.

Specify the protection code according to the standard syntax rules described in Section 2.7.5. If you do not specify **/PROTECTION**, all categories of users are allowed all types of access. If you omit a category of user when specifying protection, that category is denied all accesses.

The system applies only read and write access restrictions for magnetic tapes; create and delete are meaningless. In addition, both the system and the owner are given read and write access regardless of what you specify in the protection code.

Keywords Applicable Only to Disk:

/ACCESSED=n Specifies the number of directories to be maintained in system space for ready access. The maximum value of n is 255. Operator privilege is required to use the **/ACCESSED** keyword.

If you do not specify **/ACCESSED**, the Initialize command uses a value of 3 by default.

/BADBLOCKS=list Specifies those areas on the volume that are faulty. The Initialize command marks the areas as allocated so that no data is written in them.

You can specify one or more areas using either or both of the formats shown below. If you specify more than one area, separate specifications with a comma and enclose the list in parentheses.

lbn[:count] Specifies a logical block number on the disk volume and, optionally, a count of logical blocks beginning with the logical block specified, to be marked allocated.

MCR COMMANDS

`sector.track.cyl[:count]` Specifies a specific sector, track, and cylinder on the disk volume and, optionally, a count of blocks beginning with the first block specified to be marked allocated.

Use of the `/BADBLOCKS` keyword is device dependent. It is not required for RK06 and RM03 disks; nor is it required for disks that have been scanned for bad blocks using the BAD utility program, which is described in the VAX/VMS Operator's Guide.

`/CLUSTER_SIZE=n` Defines in blocks the default allocation unit. The maximum value of `n` is 1/100 of the volume size. The minimum value is calculated using the following formula:

$$\text{minimum} = \frac{\text{disk size}}{255 * 4096}$$

If you do not specify `/CLUSTER_SIZE`, the Initialize command uses the following values by default.

Device	Default Cluster Size
RK06	2
RK07	4
RM03	6
RP04/5	6
RP06	11

You cannot specify `/CLUSTER_SIZE` for structure level 1 volumes.

`/DATA_CHECK[=options,...]`
`/NODATA_CHECK`
(default) Defines a default for data check operations following all read and/or write operations on the volume. You can specify either or both of the following options:

READ	Performs data checks after all read operations
WRITE	Performs data checks after all write operations

If you specify `/DATA_CHECK` without specifying an option, the system assumes `/DATA_CHECK=WRITE`. By default, the system performs no data checking. You can override the checking you specify at initialization when you issue a Mount command for the volume.

You cannot specify `/DATA_CHECK` for structure level 1 volumes.

MCR COMMANDS

- /DIRECTORIES=n** Specifies the number of entries to preallocate for user directories. The maximum allowable value of n is 16000. If you do not specify /DIRECTORIES, the Initialize command uses a value of 16 by default.
- /EXTENSION=n** Specifies the number of blocks to use as a default extension size for all files on the volume. The default extension size is used when a file being updated increases to a size greater than its initial allocation. The maximum allowable value of n is 65535.
- If you do not specify a default extension size, the Initialize command uses a value of 5.
- /FILE_PROTECTION=code** Defines the file protection to be applied to all files on the volume. Specify the code according to the standard syntax rules for protection, as described in Section 2.7.5.
- This default protection is not used when the volume is being used on a VAX/VMS system. It is used on RSX-11M systems. VAX/VMS uses your default file protection.
- /GROUP** Defines a disk volume as a group volume. The owner UIC of the volume defaults to the group number of the user issuing the command and a member number of 0.
- The /GROUP keyword establishes the volume protection as RWED for system, owner, and group.
- /HEADERS=n** Specifies the number of file headers to be allocated initially for the index file. By default, the Initialize command allocates 16 file headers.
- The minimum allowable value of n is 16. The /MAXIMUM_FILES qualifier specifies the maximum.
- /INDEX=position** Requests that the index file for the volume's directory structure be placed in a specific location on the volume. You can specify one of the following options:
- | | |
|-----------|--|
| BEGINNING | Places the index file at the beginning of the volume. |
| MIDDLE | Places the index file in the middle of the volume. |
| END | Places the index file at the end of the volume. |
| BLOCK:n | Places the index file at the beginning of the logical block specified. |
- By default, the Initialize command places the index file in the middle of the volume.

MCR COMMANDS

/MAXIMUM_FILES=n Restricts the maximum number of files that the volume can contain. If you do not specify **/MAXIMUM_FILES**, Initialize uses the following values by default.

Device	Default File Maximum
RK06 Disk	4000
RK07	8000
RP04/RP05 Disk	15000
RP06 Disk	25000
RM03	15000
Floppy	123

The maximum size you can specify for any volume is the volume size in blocks divided by (cluster factor plus 1).

You cannot change the maximum file limit on a volume without reinitializing the volume.

/SHARE (default) Defines a disk volume as shareable. The protection code for the volume defaults to all types of access for all categories of user.

/NOSHARE

Specifying **/NOSHARE** sets protection to no access for group and world.

/STRUCTURE=level Indicates the structure level of the volume. If you do not specify **/STRUCTURE=2**, the Initialize command initializes the volume as a Files-11 Structure Level 1 volume, by default.

If you specify **/STRUCTURE=1**, you cannot specify **/CLUSTER_SIZE** or **DATA_CHECK**.

/SYSTEM Defines a disk volume as a system volume. The owner UIC of the volume defaults to [1,1] and default protection allows all types of access to the volume for all users.

No user privilege is required to use the **/SYSTEM** keyword; however, only users with system UICs can create directories on system volumes.

/USER NAME=string Specifies a user name of up to 12 characters to be recorded on the volume. If not specified, the Initialize command uses the user name under which you logged in.

/VERIFIED (default) Indicates whether the disk has bad block data on it. The Initialize command assumes that disks contain bad block data and uses the data to mark the bad blocks as allocated. Use **/NOVERIFIED** to request Initialize to ignore bad block data on the disk.

/WINDOWS=n Specifies the number of mapping pointers to be allocated for the file windows. When a file is opened, the file system uses the mapping pointers to access data in the file. The default number of pointers is 7; this also is the minimum allowable value of *n*. The maximum value of *n* is 80.

MCR COMMANDS

Keywords Applicable Only to Tape:

/DENSITY=n Specifies the density in bits per inch (bpi) at which the tape is to be written. You can specify a density of either 800 or 1600.

If you do not specify **/DENSITY** for a blank tape, the system uses a default density of 1600 bpi. If you do not specify a density for a previously written tape, the system uses the density at which the tape was last written.

/OVERRIDE=options,... Requests that the accessibility or the expiration date specified for the tape be overridden. One of the following keywords must be specified as an option:

ACCESSIBILITY Indicates that the accessibility specified in nonblank header 1 and volume 1 labels of the tape is to be overridden. VAX/VMS never writes in these fields.

EXPIRATION Requests the Initialize command to ignore the expiration date on a tape volume; the date is indicated by the expiration date of the first file on the volume.

You must be the owner of a tape volume or have the user privilege to override volume protection to initialize a tape that has not reached its expiration date or has a nonblank accessibility field.

If you specify both keywords enclose them in parentheses and separate them with a comma.

Examples:

> INITIALIZE/STRUCTURE=1 DBB2:MYVOL

This command initializes the disk volume on device DBB2 as a structure level 1 volume.

> @CMDFIL.CMD

```
.  
.   
.   
INI DB2:MYVOL  
.   
.   
.
```

In the sequence above, the indirect command file CMDFIL.CMD contains a command to initialize a volume. This command is in a format compatible with the RSX-11M Initvolume command.

When the command is executed, VAX/VMS maps the device name from its RSX-11M format to the VAX/VMS physical device name format, in this case DBA2.

MCR COMMANDS

Note:

- Many of the Initialize command keywords are used to maximize input/output efficiency. For information on these parameters, see the VAX/VMS System Manager's Guide.

LINK**4.18 LINK**

The Link command invokes the VAX-11 Linker to link one or more object modules into a program image and defines the execution characteristics of the image.

Format:

```
LINK [keywords] exe-file-spec [,map-file-spec][,stb-file-spec]
      = input-file-spec [keywords],...
```

keywords	/BRIEF /[NO]CROSS_REFERENCE /[NO]DEBUG[=file-spec] /FULL /INCLUDE=module-name[,...] /LIBRARY /OPTIONS /SELECTIVE_SEARCH /SHAREABLE /[NO]SYSLIB /[NO]SYSSHR /SYSTEM[=base-address] /[NO]TRACEBACK
exe-file-spec	Provides the file specification for the output image file. By default, the linker creates an executable image with a file type of EXE.

map-file-spec	Provides the file specification for the memory allocation map. The file specification can be followed by any of the keywords, /BRIEF, /FULL, or /CROSS_REFERENCE to control the contents of the map. If you do not specify any of these keywords, the map contains the following.
---------------	---

- All of the information included in the brief listing
- A list of user-defined program sections
- A list of user-defined global symbols ordered by name

If you do not provide a file specification, the Link command does not produce a memory allocation file. If you omit the file type, the Link command uses MAP as the default type.

You must use a comma to separate the exe-file-spec parameter from the map-file-spec parameter, if present.

MCR COMMANDS

stb-file-spec

Provides the file specification for a symbol table file containing symbol definitions for all symbols in the image. The symbol table file is in object module format. It can be used in subsequent linking operations to provide symbol definitions to other images.

If you omit the symbol table file specification, the Link command does not produce a symbol table file. If you omit the file type, the Link command uses STB by default.

You must use a comma to separate the map-file-spec parameter from the stb-file-spec parameter, if present.

If you want a symbol table file, but no map file, separate the exe-file-spec and stb-file-spec parameters with two commas:

```
LINK A.EXE,,A.STB=A.OBJ
```

input-file-spec

Specifies one or more input files. The input files can be object modules to be linked, libraries to be searched for external references, libraries containing modules that are to be unconditionally included, or an options description file. All input to the linker must be native VAX-11 code.

If you specify multiple input files, separate the file specifications with a comma (,) or plus sign (+). In either case, the linker creates a single image file.

If you do not specify a file type in an input file specification, the linker supplies default file types based on the nature of the file. All object modules are assumed to have a file type of OBJ.

Command Keywords:

/BRIEF

Requests the linker to produce a brief map (memory allocation) file. /BRIEF is valid only if a map file specification also is specified. A brief map file contains the following information.

- A summary of the image characteristics
- A summary of all object modules included in the image
- A summary of link-time performance statistics

You cannot specify /BRIEF and /CROSS_REFERENCE.

Specify /FULL to obtain a complete map file.

MCR COMMANDS

`/CROSS_REFERENCE`
`/NOCROSS_REFERENCE` (default) Controls whether the memory allocation listing contains a global symbol cross reference. A symbol cross reference lists each global symbol defined in the image, its value, and all modules in the image that refer to it.

You cannot specify `/CROSS_REFERENCE` and `/BRIEF`.

`/DEBUG[=file-spec]`
`/NODEBUG` (default) Controls whether the image contains the VAX-11 debugger. If the object modules contain symbol table or traceback information for the debugger, you can specify `/DEBUG` to include the information in the image as well. If the object module does not contain symbol table or traceback information, and you specify `/DEBUG`, you must use absolute hexadecimal addresses in all debugging commands.

The `/DEBUG` qualifier optionally accepts the name of an alternate, user-specified debugger. If you specify a file specification and it does not contain a file type, the linker assumes the default file type of OBJ.

`/FULL`

Requests the linker to produce a full map (memory allocation) listing. A full listing contains the following information.

- All of the information included in the brief listing
- A complete description of image sections created
- Detailed descriptions of each program section in the image file
- A list of global symbols ordered by name
- A list of global symbols ordered by value

`/SHAREABLE`

Requests the linker to produce a shareable image file rather than an executable image. Shareable images also have a default file type of EXE.

`/SYSLIB` (default)
`/NOSYSLIB`

Controls whether the default system library (`SYS$LIBRARY:STARLET.OLB`) is to be searched automatically for unresolved references. By default, the linker searches the system library when it cannot resolve references using the input files specified in the command line. If you specify `/NOSYSLIB`, the linker does not search either `SYS$LIBRARY:VMSRTL` or `SYS$LIBRARY:STARLET.OLB`.

MCR COMMANDS

`/SYSSHR (default)`
`/NOSYSSHR`

Controls whether the linker searches the default system shareable image when it cannot resolve references in the input file(s) specified.

By default, the linker searches the shareable image `SYSS$LIBRARY:VMSRTL.EXE` and then the default system library `SYSS$LIBRARY:STARLET.OLB` when it cannot resolve references using the input file(s) specified. Use the `/NOSYSSHR` qualifier to request that only the default system library be searched.

`/SYSTEM[=base-address]`

Requests the linker to produce a system image and, optionally, defines a base address for the image. A system image cannot be run using the `RUN` command; it must be bootstrapped or otherwise loaded into memory.

The base address specifies the virtual memory location at which the image is to be loaded. The address can be expressed in decimal, hexadecimal, or octal, using the radix operators `%D`, `%X`, or `%O`, respectively. If you do not specify a base address, the linker uses the default address of `%X80000000`.

If you specify `/SYSTEM`, you cannot specify `/SHAREABLE` or `/DEBUG`.

System images are intended for special purposes, such as stand-alone operating systems or diagnostics. When the linker creates a system image, it orders the program sections alphanumerically and ignores all program section attributes.

`/TRACEBACK (default)`
`/NOTRACEBACK`

Controls whether the linker includes traceback information in the image file. By default, the linker includes traceback information so that the system can trace the call stack when an error occurs. If you specify `/NOTRACEBACK`, no traceback reporting is performed when an error occurs.

If you specify `/DEBUG`, `/TRACEBACK` is also assumed.

File Keywords:

`/INCLUDE=module-name[,...]`

Indicates that the associated input file is an object module library, and that the modules specified are to be unconditionally included as input to the linker.

If the associated file specification of the library does not include a file type, the linker uses the default file type of `OLB`.

MCR COMMANDS

At least one module name must be specified. If you specify more than one name, separate them with commas and enclose the list in parentheses.

The `/LIBRARY` keyword causes conditional inclusion of object modules and can be used in combination with `/INCLUDE`.

You cannot specify a library as the first input file unless you also specify the `/INCLUDE` keyword.

`/LIBRARY`

Indicates that the associated input file is an object module library that is to be searched for modules resolving undefined symbols in the input files.

If the associated file specification of the library does not include a file type, the linker uses the default file type of `OLB`.

You can use both `/INCLUDE` and `/LIBRARY` to qualify a file specification. In this case, the explicit inclusion of modules occurs first; then the library is used to search for unresolved references.

Likewise, the same file specification can appear more than once in the command line with various combinations of `/INCLUDE` and `/LIBRARY`.

`/OPTIONS`

Indicates that the associated input file contains a list of options to control linking. If you specify `/OPTIONS` and the associated file specification does not include a file type, the linker uses the default file type of `OPT`.

The `/OPTIONS` keyword can be used to extend the length of a Link command and to provide the linker with additional link-time data.

The options file is detailed in the VAX-11 Linker Reference Manual.

`/SELECTIVE_SEARCH`

Indicates that the associated input file is an object module, and that any symbols defined in it that are not necessary to resolve outstanding references should be excluded from the symbol table of the output image file, and also from the symbol table file if `/SYMBOL_TABLE` is specified. Any binary code in the object module is always included.

Example:

> LINK MYPROG,MYPROG/FULL=MYPROG

The above command requests the linking of the object file MYPROG.OBJ. It was produced by either the VAX-11 MACRO assembler or the VAX-11 FORTRAN IV-PLUS compiler. The linker creates an executable image file and a full map file.

> LINK/SHAREABLE BETA=BETA,DELTA

The above command requests the linker to produce a shareable image file from the object modules BETA and DELTA. An options file must be used subsequently to link this shareable image with object modules to produce an executable image.

Notes:

- Use an options file to link a shareable image with object modules to produce an executable image.
- RSX-11M does not support the Link command.

Login Procedure

4.19 LOGIN PROCEDURE

VAX/VMS does not have a Login command; rather, you gain the attention of the login procedure by pressing CTRL/C, CTRL/Y, or RETURN. Login prompts for your user name and password. The login procedure is equivalent to the RSX-11M Hello command in that it verifies your right to use the system.

Format:

```
Username: user name [keywords]
Password: password
```

```
keywords          /CLI=interpreter
                  /DISK=device-name
```

```
user-name          Is the user name stored in your user authorization
                  file entry.
```

```
password           Is the password stored in your authorization file
                  entry. VAX/VMS does not display the password.
```

Command Keywords:

```
/CLI=interpreter  Specifies the name of an alternate command
                  interpreter. You can specify either the DCL or
                  MCR command interpreter. If a command interpreter
                  is not specified, your default command interpreter
                  is used.
```

```
/DISK=device-name Specifies the name of a disk device to be
                  associated with SYS$DISK for the terminal session.
                  If you do not specify /DISK, the default SYS$DISK
                  named in your authorization file is used.
```

Example:

```
<^Y>
Username: DUMPTY /CLI=MCR
Password:

                WELCOME TO VAX/VMS VERSION n.nn
>
```

CTRL/Y gets the attention of login, which prompts for the user name. The user name is entered followed by keywords requesting the MCR command interpreter. After validating the user name, login prompts for the password.

Notes:

- VAX/VMS automatically maps references to SY0 to the device associated with SYS\$DISK.
- When you login with the MCR command interpreter, it uses the login file specified in your user authorization file entry or searches your default directory for a file named LOGIN.CMD. If one is present, the MCR command interpreter executes it as an indirect command file.

LOGOUT

4.20 LOGOUT

The Logout command terminates an interactive terminal session. The system displays a termination message and performs any necessary cleanup operations, such as terminating the current image if one exists, dismounting any private volumes that remain mounted, and deallocating devices. Finally, it deletes your process and subprocesses, if any.

Format:

LOGOUT [keyword]

/FULL Requests the long form of the logout message.
 /BRIEF Requests the short form of the logout message.
 /BRIEF is the default.

Examples:

```
> LOGOUT
HUBBARD    logged out at 23-JAN-1978 17:48:56.73

> LOG/FULL
BELKNAP    logged out at 24-JAN-1978 14:23:45.30
```

```
Accounting information:
Buffered I/O count      31      Peak working set size      100
Direct I/O count        9      Peak virtual size          300
Page faults             66      Mounted volumes          2
Elapsed CPU time    0 00:00:00.33  Elapsed time          0 00:00:13.27
```

This command displays a summary of accounting statistics for the terminal session.

Notes:

- RSX-11M does not support the Logout command.
- The Bye command is supported by both VAX/VMS and RSX-11M.

MOUNT

4.21 MOUNT

The Mount command makes a disk or magnetic tape volume or volume set and the data it contains available for processing by system commands or user programs.

Format:

MOUNT [keywords] device-name,... volume-label,... [logical-name[:]]

keywords /ACCESSED=n (D)
 /BLOCK=n (T)
 /DATA_CHECK[=option,...]
 /DENSITY=n (T)
 /EXTENSION=n (D)
 /FOREIGN
 /GROUP (D)
 /[NO]LABEL (T)
 /OVERRIDE=option,...
 /OWNER_UIC=[g,m]
 /PROCESSOR=option
 /PROTECTION=code
 /RECORDS=n (T)
 /[NO]SHARE (D)
 /SYSTEM (D)
 /UNLOCK (D)
 /WINDOWS=n (D)
 /[NO]WRITE

D = applicable to disk only.

T = applicable to tape only.

Keyword descriptions are categorized accordingly below.

device-name Specifies the physical device name or logical name of the device on which the volume is to be mounted.

When mounting a volume set, you can specify more than one device name. Separate the device names with either a comma (,) or plus sign (+). If you are mounting tape volumes, you can specify more volume labels than device names.

If you specify a colon (:) following the device name, the command strips it.

volume-label Specifies the alphanumeric label written on the volume when it was initialized. For Files-11 Structure Levels 1 and 2, the volume label can have up to 12 characters; for tape volumes, it can have up to 6 characters.

If you specify more than one volume label, separate the labels with a comma (,) or plus sign (+). The volumes must be in the same volume set and the labels must be specified in the correct order.

MCR COMMANDS

The volume-label parameter is not required when you mount a volume with the /FOREIGN qualifier, nor when you specify /OVERRIDE=IDENTIFICATION. To specify a logical name when you enter either of these qualifiers, type any alphanumeric characters in the volume label parameter position.

logical-name

Defines a 1- to 63-character alphanumeric string that is the logical name to be associated with the device(s). The logical name subsequently can be used to refer to the volume or volume set in commands and programs.

If you do not specify a logical name, the Mount command assigns the default logical name DISK\$volume-label for disk devices or TAPE\$volume-label for tape devices.

The Mount command places the logical name in the process logical name table unless /GROUP or /SYSTEM is requested. In the latter cases, the name is placed in the group or system logical name tables, respectively.

If you specify multiple devices and a logical name, the logical name applies to all devices specified.

Keywords Applicable to Disk and Tape:

/DATA_CHECK[=options,...] Specifies that all read and write operations to the volume are to be followed by a write check operation. The write check compares the data on disk with that in memory to ensure that the transfer was successful. You can specify either or both of the following options.

READ Performs checks following all read operations.

WRITE Performs checks following all write operations.

If you specify /DATA_CHECK without including an option, the default of /DATA_CHECK=WRITE is used.

The data check operation is described in the VAX/VMS I/O User's Guide.

/FOREIGN

Indicates that the volume is not in the standard ANSI format used by VAX/VMS. If you do not specify /FOREIGN, the volume is assumed to be in Files-11 format.

If you specify /FOREIGN, the program that reads the volume must be able to process the labels on the volume, if any. For example, when you mount a DOS-formatted tape, you must use FLX to transfer it.

MCR COMMANDS

The default protection applied to foreign volumes is read and write for system and owner. If you specify /GROUP in addition, group members are given read and write access. If you specify /SYSTEM or /SHARE, group and world are both given read and write access. Any user category that has read and write access to a foreign volume also is allowed logical I/O and physical I/O access to the volume.

To protect a foreign volume, you must specify the /PROTECTION keyword. If the volume is currently in Files-11 format, you must have override volume protection privilege to mount it as foreign, or you must be the volume owner.

/OVERRIDE=IDENTIFICATION

Allows you to mount a volume when you do not know what the volume label is. If you specify /OVERRIDE=IDENTIFICATION, you can specify anything for the volume-label parameter; the Mount command ignores whatever you enter. The volume must be mounted /NOSHARE, either explicitly or by default.

Overriding the volume label does not affect the protection applied to the volume.

Additional override options are applicable only to magnetic tape, as described below.

/OWNER_UIC=[g,m]

Requests that the specified UIC be assigned ownership of the volume while it is mounted, thereby overriding the ownership recorded on the volume. If you are mounting a device /FOREIGN, /OWNER_UIC requests an owner UIC other than your current UIC.

The brackets are required.

You must have the privilege to override volume protection to use the /OWNER_UIC keyword, or you must be the owner of the volume.

/PROCESSOR=option

Allows you to control whether the volume has its own ACP process and which ACP image the process executes. The following options can be specified.

UNIQUE	Creates a new process to contain a copy of the default ACP image for the specified device type or controller.
--------	---

MCR COMMANDS

\SAME:device Requests that the same ACP process currently being used by the specified device be used.

file-spec Creates a new process to contain the specified ACP image (for example, a modified ACP).

Operator privilege is required to use the /PROCESSOR keyword.

/PROTECTION=code

Specifies the protection code to be assigned to the volume. All access to any file on the volume must pass both the volume protection and the protection applied to the individual file.

Specify the code according to the standard syntax rules for specifying protection as described in Section 2.7.5.

If you do not specify protection, it defaults to that assigned when the volume was initialized.

The protection specified by this keyword takes precedence over the protection implied by other keywords, for example, /SYSTEM and /GROUP. The only exception is /NOWRITE.

/WRITE (default)
/NOWRITE

Control whether the volume can be written. By default, a volume is considered read/write. Specify /NOWRITE to provide read-only access to protect files.

/NOWRITE is equivalent to writelocking a drive.

Keywords Applicable Only to Disk:

/ACCESSED=n

Specifies the approximate number of directories that will be in use concurrently on the volume.

Operator privilege is required to use /ACCESSED.

/EXTENSION=n

Specifies the number of blocks by which the file is extended. This number can be overridden by a command or program request. The minimum value of n is 0; the maximum is 65535.

If you do not specify /EXTENSION, the extension value defaults to that specified when the volume was initialized.

MCR COMMANDS

/GROUP

Makes the volume available to any user with the same UIC group number as the user issuing the Mount command. These users are not required to issue a Mount command to gain access to the volume. Volume protection is, however, applied.

The logical name for the device is placed in the group logical name table. You must have the appropriate privilege to place a name in the group logical name table.

A volume mounted /GROUP remains mounted until it is explicitly dismounted; it is not automatically dismounted when its accessors log out of the system.

If you issue a Mount command for a volume that is already mounted /GROUP, you must use the /SHARE keyword. Any additional keywords are ignored.

/SHARE /NOSHARE (default)

Indicates whether the volume is shareable. If the volume has already been mounted /SHARE by another user, and you request that it be mounted with the /SHARE keyword, any other keywords you specify are ignored.

A volume mounted /SHARE is automatically dismounted when all of its accessors log out of the system.

By default, the Mount command assumes that a device is not shareable and allocates it.

If you have previously allocated the device and then specify the /SHARE keyword, the device is deallocated.

/SYSTEM

Makes the volume available to all users of the system as long as the UIC-based volume protection allows them access. These users are not required to issue a Mount command to gain access to the volume.

The logical name for the device is placed in the system logical name table. You must have the appropriate privilege to place a name in the system logical name table.

A volume mounted /SYSTEM remains mounted until it is explicitly dismounted; it is not automatically dismounted when its accessors log out of the system.

If you issue a Mount command for a volume that is already mounted /SYSTEM, you must use the /SHARE keyword. Any additional keywords are ignored.

MCR COMMANDS

/UNLOCK Requests write access to the index file on the volume. The /UNLOCK keyword is allowed only if the volume is mounted /NOSHARE either explicitly or by default.

/WINDOWS=n Specifies the number of retrieval pointers to be used in mapping data when files are opened of the volume. If /WINDOWS is not specified, the number of pointers defaults to the value specified when the volume was initialized.

Operator privilege is required to use the /WINDOW keyword.

Keywords Applicable Only to Tape:

/BLOCK=n Sets the default block size to the specified number of bytes per block, or
/NOBLOCK requests using /NOBLOCK that each block contain one record.

By default, records are written to tape volumes in 2048-byte blocks, unless /FOREIGN and /NOLABEL are specified; then the default block size is 512 bytes.

/DENSITY=n Specifies the density at which the tape is to be written. The specified density is used only if /FOREIGN or /NOLABEL is specified and the first operation performed on the tape is a write. If density is specified, n can have either of the following values: 800 or 1600.

If density is not specified, the density defaults to 1600 bpi.

If you specify /LABEL, or if the first operation on the tape is a read, the tape is read or written at the density at which the first records on the tape is recorded.

/LABEL (default) Indicates whether the tape contains
/NOLABEL standard labels. If you mount a tape specifying the /NOLABEL keyword, an end-of-file condition is returned when a tape mark is encountered when reading the tape.

The default protection for unlabeled tapes is all access to the system and owner and no access to the group and world.

MCR COMMANDS

/OVERRIDE=option,...

Inhibits the following protection checks performed by the Mount command.

ACCESSIBILITY Indicates that the accessibility specified in the header 1 and volume 1 labels of the tape is to be overridden

EXPIRATION Allows you to write on a tape that has not yet reached its expiration date. You must have the privilege to override volume protection or you must own the volume.

SETID Allows you to inhibit the checking of the volume set identification when switching reels in a multivolume set.

IDENTIFICATION Allows you to mount a volume when you do not know that the volume label is. If you specify /OVERRIDE=IDENTIFICATION, you can specify anything for the volume-label parameter; the Mount command ignores whatever you enter.

Overriding the volume label does not affect the protection applied to the volume.

If you specify more than one keyword, enclose them in parentheses and separate them with a comma.

/RECORDS=n

Specifies the number of bytes in a record. The value of n must be no greater than the specified block size.

Examples:

```
> ASN DBAl:=DK0:
> @RODIN
MOUNT DK0:MYVOL
% MOUNT-I-MOUNTED, MYVOL mounted on _DBAl:
.
.
.
```

The Assign command defines DBAl as the equivalence name for the logical name DK0. When the indirect file RODIN.CMD executes, its command to mount the volume labeled MYVOL on DK0 becomes, in effect, a request to mount MYVOL on DBAl.

MCR COMMANDS

```
> MOUNT DBA1 MYVOL DISK
% MOUNT-I-MOUNTED, MYVOL mounted on _DBA1:
```

This command mounts the volume MYVOL on disk device DBA1 and assigns a logical name of DISK.

Notes:

- The form of the Mount command that is compatible between RSX-11M and VAX/VMS is as follows.

MOU llmn:volume-label

llmn is an RSX-11M physical device name that has been assigned as the logical name for a VAX/VMS physical device unit.

volume-label is the volume label.

- VAX/VMS and RSX-11M do not support the same keywords for the Mount command.
- VAX/VMS syntax for the Mount command indicates a space between the device name and the volume label. VAX/VMS accepts either a colon or a space as a delimiter between the device name and volume label.

MCR COMMANDS

4.22 ON STATEMENT

You can use the On statement to control the following:

- Action taken when a program, command, or command procedure returns an error status
- Action taken when CTRL/Y is pressed.

RSX-11M does not support either form of the On statement.

ON CONTROL_Y

4.22.1 On Control Y

The On Control Y statement allows you to specify an action to be taken when CTRL/Y is pressed and is normally used only within indirect command files. Once the On Control Y statement has been processed, the command interpreter reacts to the pressing of CTRL/Y in either of the following ways:

- If an image is active when you press CTRL/Y, an image exit occurs, thus allowing termination handlers, if any, to run; then the specified action is taken.
- If no image is active (that is, the command interpreter is executing), the specified action is taken when the command completes.

Once an On Control Y action has been established, it remains until it is replaced by a subsequent On Control Y statement or canceled by a Set NoControl Y command. It does not have to be re-established each time CTRL/Y is pressed.

If one level of an indirect command file issues an On Control Y statement and then invokes another level, the default On Control Y action established for the nested level is as follows:

```
ON CONTROL_Y THEN EXIT
```

Thus, if you press CTRL/Y while the nested level is executing, it causes a return to the level from which it was invoked. Pressing CTRL/Y again causes the On Control Y action established by the first level to be taken.

Format:

```
ON CONTROL_Y THEN command
```

command specifies the action to be taken. It can be any valid VAX/VMS MCR command.

Example:

```
> @APPLIC.CMD
  ON CONTROL_Y THEN LOGOUT
  .
  .
  .
  Y
```

Once the indirect command file APPLIC.CMD is started, the user can interact only with the application. Because the On Control Y statement indicates that the command interpreter is to effect a logout when CTRL/Y is pressed, the user of the application cannot return to use the command interpreter.

MCR COMMANDS

Notes:

- The VAX/VMS System Services Reference Manual describes exit handlers.
- If you wish to run an image that should not be interrupted by CTRL/Y, issue a Set NoControl Y command prior to initiating the image. When the image exits, you can re-enable CTRL/Y handling by issuing the following commands.

```
ON CONTROL_Y THEN command  
SET CONTROL_Y
```

ON severity-level

4.22.2 On Severity-level

The On severity-level statement defines the default course of action to be taken when a command or program executed within an indirect command file encounters an error condition.

Return status values in VAX/VMS include a severity-level code. The code groups status returns to indicate success, warning, error, or severe error. When a command or program is executed, the return status value is compared with the current severity level to determine subsequent action.

The On severity-level statement is used with the Set On command. By default, the initial state for an indirect command file is to have condition recognition disabled (Set NoOn). You can enable condition recognition by including a Set On command or an ON statement in the indirect command file.

Format:

ON	severity-level	THEN	command
severity-level	Specifies the severity of the condition that is to cause the indicated action to be taken. The severity level is represented by one of the following keywords.		
	WARNING	The specified action is to be taken for warning, error, and severe error returns.	
	ERROR	The specified action is to be taken for error and severe error returns; the default action for warnings is to continue.	
	SEVERE_ERROR	The specified action is to be taken only for severe errors; the default action for warnings and errors is to continue.	
	You can truncate these keywords to one or more characters.		
command	Specifies the action to be taken when errors equal to or greater than the specified level of error occur. You can specify any valid MCR command line following the keyword THEN.		
	If you do not include an On statement in an indirect command file and a Set On command has been executed, the command interpreter issues an EXIT command when errors or severe errors result from the execution of a command or program. In all other cases, command execution continues.		
	Once the command in an On statement has been executed as a result of an error condition, the default action, Exit, is reinstated. It remains in effect until a new On command is encountered.		

Example:

```

> @INDFILE
  RUN POMME
    .
    .
    .
  ON ERROR THEN CONTINUE
  RUN SQUASH
  RUN PEAS
    .
    .
    .
  ON SEVERE_ERROR THEN .GOTO ERR
  RUN TOMATO
  RUN PEPPER

.ERR: RUN MARKET
EXIT

```

When execution of the indirect command file starts, any error more severe than a warning causes the command interpreter to issue an Exit command by default.

Once the first On statement is encountered, the indirect command file can continue after any type of error. It continues from warnings by default and from errors and severe errors as a result of the Continue command in the On statement.

Once the second On statement is encountered, the indirect command file is allowed to continue from warnings and errors, but severe errors cause a branch to the portion of the indirect command file labeled ERR.

Notes:

- Once the clause (THEN command) of a condition statement is executed, condition recognition is set to ON ERROR THEN EXIT.
- Any On severity-level statement in an indirect command file overrides the effect of a previous On statement.
- You can control whether the command interpreter checks the status returned from commands with the Set On command.
- You can use the Set NoOn command to temporarily disable condition recognition.

PRINT

4.23 PRINT

The Print command queues one or more files for printing on either a system-defined printer or a device that you specify. The system considers a file or files queued by a Print command to be a job. It assigns a unique job identification to each job in the system.

Format

PRINT [keywords] file-spec[keywords],...

keywords	/AFTER=absolute-time
	/BURST
	/COPIES=n
	/DELETE
	/DEVICE=device-name
	/[NO] FEED
	/[NO] FLAG_PAGE
	/FORMS=type
	/[NO] HEADER
	/[NO] HOLD
	/[NO] IDENTIFY
	/JOB_COUNT=n
	/[NO] LOWERCASE
	/NAME=job-name
	/PAGE_COUNT=n
	/PRIORITY=n
	/QUEUE=queue-name
	/SPACE [=n]

file-spec	Provides the file specification of one or more files to be printed. If you specify multiple files, separate them with either a comma (,) or a plus sign (+).
-----------	--

You can use wild cards in place of the directory, file name, type, or version fields of the file specification.

If you omit a file type, the Print command uses LIS by default.

Command Keywords:

/AFTER=absolute-time	Requests that the job be printed after a specific time of day. The time must be specified in 24-hour format as follows:
----------------------	---

hh[:mm]

For example /AFT=17 places a job in a hold status until 5:00 pm. If the time already has passed, the job is printed immediately.

/DEVICE=device-name	Requests that the files specified be queued to a particular device. If you do not specify /DEVICE, files are queued to the current default printer. The default device is SYS\$PRINT.
---------------------	---

MCR COMMANDS

/FORMS=type	Specifies the form type required for the files to be printed. Specify the form type using a numeric value or alphanumeric code. Codes for form types are installation defined.
/HOLD /NOHOLD (default)	Requests that the file(s) be queued but not printed. The file cannot be released for printing until you issue the Set Queue command to release it.
/IDENTIFY (default) /NOIDENTIFY	Controls whether the Print command displays a message indicating the job identification of the print job and the name of the device to which it has been queued. By default, the Print command displays this information in the following format. JOB jobid ENTERED ON QUEUE device-name
/JOB_COUNT=n	Requests that the entire job be printed n times. The value of n is a decimal number ranging from 1 through 255. If you omit this keyword, one copy of the job is printed.
/LOWERCASE /NOLOWERCASE (default)	Indicates that the specified file(s) contain lowercase alphabetic letters and must be printed on a printer that has uppercase and lowercase capabilities.
/NAME=job-name	Defines a 1- to 8-character alphanumeric string to identify the job. The name string is displayed in response to the Show Queue command. If name is not specified, the name string defaults to the file name of the first file in the job.
/PRIORITY=n	Specifies the priority of the print job. The value of n must be in the range of 0 through 31; 0 is the lowest priority, and 31 is the highest. By default, jobs are assigned the same priority as your process. You can assign lower priorities to your nonessential jobs so that jobs you want sooner will be processed first. Privilege is required to set a priority value that is higher than your process's priority.
/QUEUE=queue-name	Requests that the file(s) specified be queued to a particular device. The default is SYS\$PRINT. This keyword performs the same function as /DEVICE.

MCR COMMANDS

File Keywords:

<code>/BURST</code> <code>/NOBURST</code>	Indicates whether a burst page is to precede the file. A burst page is printed before a flag page and contains the same information; however, it is printed over the perforation between pages to make the separation of files easier. This keyword overrides the installation's default established for printers when they are initialized for spooling. <code>/NOBURST</code> is the default for user-specified devices.
<code>/COPIES=nn</code>	Specifies the number of copies of the file to be printed. The value of nn is a decimal number in the range of 1 through 255. The default number of copies is 1. If you specify <code>/COPIES</code> as a command keyword, each file named in the command is printed the specified number of times.
<code>/DELETE</code> <code>/NODELETE (default)</code>	Indicates whether the file is to be deleted after printing. If you specify this keyword as a command keyword, all files specified are deleted.
<code>/FEED (default)</code> <code>/NOFEED</code>	Controls whether the Print command automatically inserts form feeds when it prints files that do not have carriage control characters. By default, the Print command inserts a form feed when the forms are within four lines of the end of the form. On standard 66-line forms, a form feed occurs after printing 62 lines.
<code>/FLAG_PAGE</code> <code>/NOFLAG_PAGE (default)</code>	Indicates whether a flag page is to be printed preceding the file. If you specify <code>/FLAG</code> with a file specification, a separate flag page is printed preceding the associated file. The flag page lists the file name. If you specify <code>/FLAG</code> with the command verb, a flag page is printed for each file in the job; the flag page lists the job name.
<code>/HEADER</code> <code>/NOHEADER (default)</code>	Controls whether the name of the file is printed at the top of each printed page.
<code>/PAGE_COUNT</code>	Specifies the number of pages of the file to be printed. <code>/PAGE COUNT</code> can be used only as a file keyword; it cannot be used as a command keyword. If this keyword is omitted, the entire file is printed.

MCR COMMANDS

`/SPACES [=n]`

Specifies the number of spaces to be left between lines of output in the specified file. The value of n indicates the number of spaces. It can be a 1 or a 2.

If `/SPACES` is omitted, no extra spaces are printed between lines of the file.

If `/SPACES` is specified without a value for n, one extra space is provided between files of output; that is, output is double spaced.

Example:

```
> PRINT/COPIES=5 THISFILE.DAT/SPACE=2,THATFILE/PAGE_COUNT=4
```

This command requests the printing of five copies of THISFILE and THATFILE. THISFILE is to be double spaced. Only the first four pages of THATFILE are to be printed.

Notes:

- RSX-11M does not support the Print command.
- You can delete a queue entry using the Delete command with the `/ENTRY` keyword.
- You can control the attributes of a print queue using the Set Queue command.

PURGE

4.24 PURGE

The Purge command deletes all but the highest numbered version or versions of a specified file or files.

Format:

PURGE [keyword] file-spec,...

keywords /KEEP=n
 /LOG

file-spec Provides the file specification of the files to be purged. If you specify more than one file, separate them with either a comma (,) or a plus sign (+). Do not specify a version number.

You can use wild cards in place of the file name or file type fields.

If you do not provide the file name and type, the Purge command purges all files in the directory.

Command Keyword:

/KEEP=n Specifies the maximum number of versions of the specified files to be retained. If /KEEP is not specified, the command deletes all but the highest numbered version.

/LOG Specifies that the names of files be displayed as they are deleted.

Examples:

> PURGE /KEEP=2 *.DAT

This command requests that all files in the default directory having a file type of DAT be purged so that only the two highest numbered versions remain.

> PURGE [122020.MSG]

This command requests the purging of all files in subdirectory [122020.MSG] to one version.

Notes:

- RSX-11M does not support the Purge command.
- You can use the /PU switch with PIP to purge files.
- Purge differs from the operation of PIP/PU in that the /KEEP keyword specifies the actual number of files to be kept rather than the highest version numbers.

RUN**4.25 RUN**

The Run command requests execution of a native or RSX-11M image. You can request an image for execution in your process (the requesting process), or you can use the Run command to create a subprocess or detached process to execute the image. By default, the image executes in the requesting process.

If you specify any keywords except /DEBUG with the Run command, the command creates a separate process to execute the specified image. The command displays the created process's identification on SYS\$OUTPUT. By default, it creates a subprocess with the same UIC, privileges, and priority as the requesting process and deducts resource quotas from the requesting process to assign them to the subprocess.

The /UIC keyword requests the Run command to create a detached process; you must have the appropriate privilege to create a detached process.

Additional keywords allow you to schedule the process, describe its resources and privileges, and indicate whether the process is to hibernate.

NOTE

The notes at the end of this command description contain important information. You should read them if you intend to use any Run command keywords.

Format:

RUN [keywords] file-spec

keywords	/[NO] ACCOUNTING /AST_LIMIT=limit /[no] authorize /BUFFER_LIMIT=limit /[NO] DEBUG /DELAY=delta-time /ERROR=equivalence-name /FILE_LIMIT=limit /INPUT=equivalence-name /INTERVAL=delta-time /IO_BUFFERED=limit /IO_DIRECT=limit /MAILBOX=mailbox-unit /MAXIMUM_WORKING_SET=limit /OUTPUT=equivalence-name /PAGE_FILE=limit /PRIORITY=n /PRIVILEGES=privilege-list /PROCESS_NAME=process-name /QUEUE_LIMIT=limit /[NO] RESOURCE_WAIT /SCHEDULE=absolute-time
----------	---

MCR COMMANDS

```
/[NO]SERVICE_FAILURE  
/SUBPROCESS_LIMIT=limit  
/[NO]SWAPPING  
/UIC=[g,m]  
/WORKING_SET=default
```

file-spec Provides the file specification of the native or RSX-11M image to be executed. If the file specification does not include a file type, the Run command uses EXE by default.

Command Keywords:

/ACCOUNTING (default) Allows or disallows the created process to specify, for any processes that it subsequently creates, that the created process can disable accounting.
/NOACCOUNTING

To use this keyword, you must have the user privilege to disable accounting.

/AST_LIMIT=limit Specifies the AST queue limit for:

- The number of ASTs that the process can have outstanding at one time
- The number of scheduled wake up requests that the process can have outstanding at one time

If you do not specify **/AST_LIMIT**, the process has a limit of 10 by default.

The minimum value of n is 2.

This limit is not deductible; see the notes below.

/AUTHORIZE (default) Determines whether login is to check the user authorization file. This keyword is meaningful only when the image to be run is LOGINOUT.EXE.
/NOAUTHORIZE

If you specify **/AUTHORIZE**, login performs all the operations that it performs when you log into the system at a terminal.

If you specify **/NOAUTHORIZE**, login bypasses the user authorization file checks.

/BUFFER_LIMIT=limit Specifies in bytes the maximum amount of memory that the created process can use for buffered I/O operations or temporary mailboxes. Mailboxes are used to emulate RSX-11M send/receive directives; see the notes below.

If you do not specify a buffered I/O limit, the default value of 4096 bytes is used; the minimum amount required for any process to execute is 1024 bytes. This limit is deductible; see the notes below.

MCR COMMANDS

/DEBUG
/NODEBUG

For a native image, controls whether the image is to be run with the debugger. If the image was linked with the VAX-11 Symbolic Debugger and you do not want the debugger to prompt, use the /NODEBUG keyword. If the image was linked with traceback, traceback reporting is performed when an error occurs.

This keyword is not meaningful for RSX-11M images built with ODT; ODT always prompts when you run an image built with it.

If you specify /DEBUG, do not specify any other Run command keywords.

/DELAY=delta-time

Requests that the created process be initialized and placed in hibernation and awakened after a specified time interval has elapsed.

If you specify /INTERVAL with /DELAY, the first wake request is scheduled for the delay time specified and all subsequent wake requests are scheduled according to the specified time interval.

/ERROR=equivalence-name

Defines a 1- to 63-character equivalence name string for the logical device name SYS\$ERROR. The logical name and equivalence name are placed in the process logical name table for the created process.

/FILE_LIMIT=limit

Specifies the maximum number of files that a process can have open at one time.

If you do not specify an open file limit for a created process, the system uses the default value of 20. The minimum amount required for a process to execute is 2.

This limit is deductible; see the notes below.

/INPUT=equivalence-name

Defines a 1- to 63-character equivalence name string for the logical device name SYS\$INPUT. The logical name and equivalence name are placed in the process logical name table for the created process.

/INTERVAL=delta-time

Requests that the created process be initialized, placed in hibernation, and awakened at regularly scheduled intervals as specified by the value of delta time.

MCR COMMANDS

If you specify /DELAY or /SCHEDULE with /INTERVAL, the first wake request occurs at the delay time or scheduled time specified. All subsequent wake requests occur at the specified interval time. If neither /DELAY nor /SCHEDULE is specified, the first wake request occurs immediately.

/IO_BUFFERED=limit

Specifies the limit on the number of outstanding buffered I/O operations permitted for the process. A buffered I/O operation is one in which the data transfer takes place from an intermediate buffer in the system pool, not from a process-specified buffer.

If you do not specify /IO_BUFFERED, the process has a buffered I/O limit of 6 by default.

The minimum value of n is 2.

This limit is not deductible; see the notes below.

/IO_DIRECT=limit

Specifies the limit on the number of outstanding direct I/O operations permitted for the process. A direct I/O operation is an I/O operation in which the data transfer takes place directly from a process-specified buffer.

If you do not specify /IO_DIRECT, the process has a limit of 6 by default.

The minimum value of n is 2.

This limit is not deductible; see the notes below.

/MAILBOX=mailbox-unit

Specifies the unit number of a mailbox to receive a termination message when the created process is deleted.

/MAXIMUM_WORKING_SET=limit

Specifies the maximum size to which the image to be executed in the process can increase its working set size. This keyword is applicable only when running a native image.

The default working set size is 150 pages. The minimum working set size required for a process to execute is 50 pages.

This limit is not deductible; see the notes below.

/OUTPUT=equivalence-name

Defines a 1- to 63-character equivalence name string for the logical device name SYS\$OUTPUT. The logical name and equivalence name are placed in the process logical name table for the created process.

MCR COMMANDS

`/PAGE FILE=limit`

Specifies the maximum number of pages that can be allocated in the paging file for the process.

If you do not specify a paging file limit, the system uses a default value of 40 256-page blocks; that is 10000 pages. The minimum number of pages required for a process to execute is one 256-page block.

This limit is deductible; see the notes below.

`/PRIORITY=nn`

Specifies the base priority at which the created process is to be executed. VAX/VMS priorities range from 0 through 31 (decimal). Priorities 0 through 15 are normal process priorities. Priorities 16 through 31 are time-critical (real-time) priorities.

You must have the appropriate privilege to set the created process's base priority to a value higher than your process's base priority. If you specify no priority or a higher priority than allowed, the system uses your process's priority by default.

`/PRIVILEGE=privilege-list`

Defines privileges for the created process. The privilege list consists of one or more of the privileges listed in Appendix A. You must have a user privilege to give a process you create any privileges that you do not have.

If you specify multiple privileges, they must be separated by commas, and the list must be enclosed in parentheses.

Any of the privileges listed can be preceded by NO to deny the process that privilege, for example, NOSWAP_MODE.

Rather than specifying individual privileges, you can specify either ALLPRIV or SAME. ALLPRIV indicates that the created process is to have all privileges. SAME indicates that the created process is to have the same privileges as your process. If you specify /PRIVILEGE=NOSAME, the created process has no privileges.

If you do not specify /PRIVILEGE, the created process has the same privileges as your process; that is, the default is PRIVILEGE=SAME.

MCR COMMANDS

<code>/PROCESS=process-name</code>	Defines a 1- to 15-character alphanumeric name for the created process. The process name is implicitly qualified by the group number of the process's UIC; that is, process names must be unique only within the group.
<code>/QUEUE_LIMIT=quota</code>	Specifies the maximum number of timer queue entries that the created process can have outstanding at any one time. This limit includes timer requests and scheduled wakeup requests. Timer requests are used to emulate RSX-11M MARK TIME directives. If you do not specify a timer queue entry quota, the system uses a default value of 10. A process does not require any timer queue limit to execute. This limit is deductible; see the notes below.
<code>/RESOURCE_WAIT (default) /NORESOURCE_WAIT</code>	Enables resource wait mode for the created process. By default, if a system resource is required for a process to execute a particular function and the resource is not available, the system places the process in a wait state until the resource becomes available. If you specify <code>/NORESOURCE_WAIT</code>, the process receives an error status code when the resource is not available. <code>/RESOURCE_WAIT</code> is the default.
<code>/SCHEDULE=absolute-time</code>	Requests that the created process be initialized, placed in hibernation, and awakened at the time of day specified by the value of absolute time.
<code>/SERVICE_FAILURE /NOSERVICE_FAILURE (default)</code>	Enables system service failure exception mode for the created process. By default, if an error occurs when a process calls a system service either directly or as a result of RSX-11M directive emulation, a status code indicating the error is returned. If you specify <code>/SERVICE_FAILURE</code> and an error occurs during the processing of a system service request, the process receives an exception condition. Native images can declare condition handlers for such exception conditions.

MCR COMMANDS

`/SUBPROCESS_LIMIT=quota` Specifies the maximum number of subprocesses that the created process is allowed to create.

If you do not specify a subprocess limit, the system uses the default value of 8. A process does not require any subprocess quota to execute.

This limit is deductible; see the notes below.

`/SWAPPING (default)`
`/NOSWAPPING`

Enables swapping for the created process. By default, a process is swapped from physical memory to allow other processes to execute. Swapping is similar in function to RSX-11M checkpointing.

You must have the appropriate privilege to specify `/NOSWAPPING`.

`/UIC=[g,m]`

Defines the user identification code for the created process. The values of g and m are numeric strings in the range of 0 through 377. The number is interpreted in octal radix.

`/WORKING_SET=default`

Specifies the default working set size for the created process; that is, the number of pages in the working set for the image to be executed.

If you do not specify a default working set size, the system uses the default value of 150 pages. The minimum number of pages required for a process to execute is 50 pages. The value specified cannot be greater than the working set quota specified by the `/MAXIMUM_WORKING_SET` keyword.

This quota is not deductible.

Examples:

> RUN RSXIMAGE.TSK

This command activates the RSX-11M image RSXIMAGE in the requesting process; no subprocess or detached process is created. The system does not prompt for subsequent commands until the image terminates.

> RUN VAXIMAGE /UIC=[122,20]

This command creates a detached process that runs under UIC [122,20] and has the same privileges and resource quotas as its requester.

MCR COMMANDS

```
> RUN FIRST.TSK /PROCESS=PAY1 /DELAY=:10  
> RUN SECOND.TSK /PROCESS=PAY2
```

The commands above create two subprocesses named PAY1 and PAY2, each of which is to execute an RSX-11M image. PAY1 hibernates as a result of the /DELAY keyword on its Run command; PAY2 is activated immediately. The two processes cooperate. SECOND, the image executing in PAY2, awakens PAY1 by issuing a directive.

The process requesting that the two images be run remains free to accept subsequent user commands.

Both subprocesses have the same privileges as their requesting process. Resource quotas are subtracted from the requesting process's.

Notes:

- Specifying a process name is not sufficient to allow emulation of the RSX-11M functions (for example, common event flag clusters and send/receive) that require a task name in the image label block.
- If you run an RSX-11M image that has a task name starting with an alphabetic character in the image label block and specify /PROCESS, the task name becomes the process name; that is, it overrides the name specified by /PROCESS.
- Mailboxes are virtual devices that can be used by both RSX-11M and native images. VAX/VMS uses mailboxes to emulate RSX-11M send/receive directives. For further information on mailboxes, see the VAX-11/RSX-11M Programmer's Reference Manual for RSX-11M images or the VAX/VMS System Services Reference Manual for native images.
- For RSX-11M images issuing send/receive directives, the mailboxes used in emulating these directives are not created until just prior to actual image execution. Therefore, using either the /INTERVAL or /DELAY keywords does not cause creation of the mailboxes when the command is issued; you cannot assume that the mailboxes exist until image execution begins.
- For an RSX-11M image, the /INTERVAL keyword has the same effect as /DELAY; that is, the process execution occurs at the designated time, and when the image exits, the process is deleted. The image is executed only once.
- For a native image, /INTERVAL causes the image to hibernate and be re-executed after the specified interval if the image issues a RET instruction. The image exits, however, if it issues an Exit system service.
- The form of the Run command that is compatible between RSX-11M systems and VAX/VMS systems is:

RUN file-spec

MCR COMMANDS

- Use the following keywords to assign equivalence names for the logical names SYS\$INPUT (TI), SYS\$OUTPUT (TO), and SYS\$ERROR (CL).

```
/INPUT
/OUTPUT
/ERROR
```

- Use the following keywords to override the default attributes for a process:

```
/ACCOUNTING
/PRIORITY
/PRIVILEGES
/PROCESS_NAME
/SERVICE_FAILURE
/SWAPPING
```

- When you issue a Run command to create a process, you can define limits to restrict the amount of various system resources available to that process. The following resource limits are deductible when you create a subprocess; that is, the values you specify are deducted from your current limits and given to the subprocess.

```
/BUFFER_LIMIT
/FILE_LIMIT
/PAGE_FILE
/QUEUE_LIMIT
/SUBPROCESS_LIMIT
```

The limit amounts are returned to you when the subprocess is deleted.

- The system defines minimum values for each specifiable limit; if you specify a limit that is below the minimum, or if you specify a deductible quota that reduces your quota below the minimum, the Run command cannot create the process. To determine your current quotas, issue the SHOW PROCESS/QUOTAS command.
- You also can specify limits that affect performance but are not deducted from your process's resources. The nondeductible limits are:

```
/AST_LIMIT
/MAXIMUM_WORKING_SET
/IO_BUFFERED
/IO_DIRECT
/WORKING_SET
```

- Hibernation is the VAX/VMS concept that corresponds to the RSX-11M concept of a suspended task.

MCR COMMANDS

- Use the following keywords to schedule execution of the image:

```
/DELAY  
/INTERVAL  
/SCHEDULE
```

If you specify any of these keywords, the Run command creates the process, places it in a state of hibernation, and schedules a wake request for the appropriate time. The process cannot execute the image until it is wakened. For RSX-11M images, /INTERVAL has the same effect as /DELAY.

- Issue a Stop command to terminate execution of the image in the process and, if the process is not your current process, cause the process to be deleted.
- Issue a Cancel command to cancel wake requests that are scheduled for the process but not yet delivered.
- When a command accepts a keyword that specifies a time value, the time value is either an absolute time or a delta time. The syntax rules for specifying time values are described in Section 2.7.6 and summarized below.

-- Absolute times have the format:

```
[dd-mmm-yyyy[:]][hh:mm:ss.ss]
```

-- Delta times have the format:

```
[dd-] [hh:mm:ss.ss]
```

4.26 SET

The Set command provides a number of options that allow you to set characteristics and defaults for your process, job, and certain devices. Table 4-1 lists the Set options.

The various Set options are described in detail in the VAX/VMS Command Language User's Guide and in the VAX/VMS Operator's Guide, as indicated in Table 4-1. The commands described in the VAX/VMS Command Language User's Guide also are summarized in the following sections of this manual. The only exception is the SET /UIC command, which is detailed in this manual.

Table 4-1
Set Command Options

Option	Function
ACCOUNTING ¹	Initializes the accounting log file
CARD_READER	Defines the default ASCII translation mode for a card reader
[NO]CONTROL_Y	Controls whether pressing CTRL/Y passes control to the command interpreter.
DEFAULT	Establishes a disk and/or directory as the current default for file specifications
DEVICE ¹	Defines device characteristics
LOGINS ¹	Allows or disallows users to log into the system
MAGTAPE	Defines characteristics of a magnetic tape device
[NO]ON	Controls whether the command interpreter checks for an error condition following the execution of a command in a command procedure
PRINTER ¹	Defines characteristics of a printer
PROCESS	Defines execution characteristics of the current process
PROTECTION	Defines the protection status of a file or group of files, or establishes the default protection to be applied to all files subsequently created during the job

¹ This command is described in the VAX/VMS Operator's Guide.

(continued on next page)

MCR COMMANDS

Table 4-1 (Cont.)
Set Command Options

Option	Function
QUEUE	Changes the attributes associated with an entry in a printer or batch job queue
RMS DEFAULT	Provides default multiblock and multibuffer count values to be used by VAX-11 RMS for file operations
TERMINAL	Defines operational characteristics of a terminal
/UIC	Changes the UIC of the current process
[NO]VERIFY	Controls whether the command interpreter displays lines in command procedures as it executes them
WORKING_SET	Changes the current working set limit or quota

SET CARD__READER

4.26.1 Set Card Reader

The Set Card Reader command defines the translation mode for a card reader.

Format:

```
SET CARD_READER device-name [keywords]
```

```
keywords      /026  
              /029
```

Example:

```
> SET CARD_READER CRA0: /029
```

This command establishes the default mode for card reader input as 029.

SET CONTROL_Y

4.26.2 Set Control Y

The Set Control Y command determines whether the command interpreter receives control when you press CTRL/Y. The CTRL/Y function key provides a general purpose escape; you can press it at any time during an interactive terminal session to interrupt the current command, command procedure, or program image.

The Set NoControl Y command is provided for use in special applications. When the Set NoControl Y command is executed in a system-specific command procedure for a particular user at login, that user can communicate only with the application program that controls the terminal.

The effect of Set NoControl Y applies to the CTRL/C function as well for all commands and programs that do not have special action routines to respond to CTRL/C.

Format:

```
SET CONTROL_Y
```

Example:

```
> SET NOCONTROL_Y
```

After the system executes this command, the CTRL/Y function is disabled.

Note:

- If the user authorization file contains a file name of a system-specific indirect command file, your LOGIN.CMD file is not executed when you log into the system.

SET DEFAULT

4.26.3 Set Default

The Set Default command establishes a new default device and/or directory for the current terminal session. The new defaults are applied to all subsequent file specifications that do not explicitly name the device or directory.

Format:

```
SET DEFAULT [device-name[:]][<directory>]
```

Examples:

```
> SHOW DEFAULT
DBB1:[MONET]
> SET DEFAULT [RENOIR]
```

The Show Default command displays the name of the default disk device as DBB1 and the directory as [MONET]. The Set Default command establishes [RENOIR] as the new default directory. DBB1 remains as the default disk.

```
> SET DEFAULT DBB2:[HASSAM]
```

This command establishes DBB2 as the default disk device and [HASSAM] as the default directory name.

Note:

- When a new default disk device is established, the system equates it with the logical name SYS\$DISK. SYS\$DISK is like SY0 in that it is applied by default.

SET MAGTAPE

4.26.4 Set Magtape

The Set Magtape command defines the characteristics associated with a specific magnetic tape device for all subsequent file operations. The tape must be unmounted or mounted /FOREIGN.

Format:

```
SET MAGTAPE device-name [keywords]
```

```
keywords      /DENSITY=density  
              /REWIND  
              /UNLOAD
```

Example:

```
> SET MAGTAPE MTB2:/DENSITY=800
```

This command sets the default density for writing files using MTB2 to 800 bpi.

SET ON**4.26.5 Set On**

The Set On command requests that the command interpreter perform error checking following the execution of commands in an indirect command file; Set NoOn inhibits error checking. During the execution of indirect command files, the command interpreter checks the status code returned when an MCR command or directive or program image completes; it saves the numeric value of the code in the special variable \$STATUS. The low-order three bits of this value also are saved in the special symbol \$SEVERITY.

The On command, described in Section 4.22, establishes an action to be taken based on the value of the symbol \$SEVERITY; the default action is to stop the execution of the indirect command file when errors or severe errors occur.

Format:

```
SET [NO]ON
```

Example:

```
SET NOON
DELETE MEANS.*;4
SET ON
.
.
.
```

The Set NoOn inhibits error checking at the end of a command. As a result, no errors are reported during the delete operation. The subsequent Set On command re-enables error checking.

Notes:

- When Set NoOn is in effect, the command interpreter continues to place the status code value in the symbol \$STATUS and the severity level in \$SEVERITY; it does not, however, perform any action based on the value.
- The Set NoOn command applies only within the indirect command file in which it is executed. If you use the Set NoOn command in one indirect command file that invokes another indirect command file, the default condition of Set On is established automatically for the second file.

SET PROCESS

4.26.6 Set Process

The Set Process command lets you change the priority of a process and enable or disable resource wait mode or swapping for a process. You can specify the process with either a process name or a process identification. If you specify neither, the characteristics of the current process are changed by default.

Format:

```
SET PROCESS keywords [process-name]
```

```
keywords    /IDENTIFICATION=pid  
            /PRIORITY=n  
            /[NO]RESOURCE_WAIT  
            /[NO]SWAPPING
```

Example:

```
> SET PROCESS/PRIORITY=7/NOSWAPPING
```

This command changes the current process's priority to 7 and disables swapping for the process. You must have the appropriate privilege to raise your priority higher than the default priority contained in your user authorization file entry. You also must have the appropriate privilege to disable swapping.

SET PROTECTION

4.26.7 Set Protection

The Set Protection command establishes the protection that is applied to a file or group of files, or it establishes the default protection for all files subsequently created during the terminal session. You can specify protection types of read (R), write (W), execute (E), and delete (D) for the categories system, owner, group, and world.

Format to Set Protection on Files:

```
SET PROTECTION=[([SYSTEM:prot,] [OWNER:prot,] [GROUP:prot,]
                [WORLD:prot])] file-spec,...[/PROTECTION=...],...
```

Format to Set Default Protection:

```
SET PROTECTION=([SYSTEM:prot] [OWNER:prot] [GROUP:prot] [WORLD:prot]/DEFAULT
```

Examples:

```
> SET PROTECTION=(SYSTEM:RWED,OWNER:RWED,WORLD:R) -
> _TREEN.DAT /PROTECTION=(GROUP:RW),-
> _CHARGER.EXE /PROTECTION=(GROUP:RE)
```

This command first establishes the protection for both files as read, write, execute, and delete for the system and owner and as read-only for the world. It then specifies that, for the file TREEN.DAT, group members are allowed read and write access and that, for file CHARGER.EXE, group members are allowed read and execute access.

```
> SHOW PROTECTION
SYSTEM=RWED, OWNER=RWED, GROUP=RWED, WORLD=R
> EDI HEREZ.FOR
[CREATING NEW FILE]
INPUT
.
.
.
> SET PROTECTION=(GROUP:R,WORLD) HEREZ.FOR
```

The Show Protection command displays the default file protection for the current process. When the file HEREZ.FOR is created, the default protection is given to it. The Set Protection command then modifies the protection for HEREZ.FOR as follows:

- Protection for system and owner remain unchanged
- Group access is restricted to read-only
- The world is denied access to the file

```
> SET PROTECTION=(SYSTEM:RWED,OWNER:RWED,GROUP:RE,WORLD:R)/DEFAULT
```

This command changes the default file protection for the duration of the terminal session.

SET QUEUE

4.26.8 Set Queue

The Set Queue command changes the status or attributes of a file that is queued for printing or for batch job execution but that has not yet been processed by the system. The system assigns a unique entry number, called a job identification, to each queued printer or batch job in the system. The Print and Submit commands display the job identification when they successfully queue a job for processing.

Format:

```
SET QUEUE/ENTRY=jobid [keywords] [queue-name]
```

```
keywords      /AFTER=absolute-time
               /FORMS=type
               /HOLD
               /JOB COUNT=n
               /[NO]LOWERCASE
               /NAME=job-name
               /PRIORITY=n
               /RELEASE
```

Examples:

```
> PRINT/HOLD RAH.DAT
Job 134 entered on queue SYS$PRINT
.
.
.
> SET QUEUE/ENTRY=134/RELEASE
```

The Print command requests that the file RAH.DAT be queued to the system printer and placed in a hold status. The Set Queue command releases the file for printing.

```
> SUBMIT MYJOB.COM
job 210 entered on queue SYS$BATCH
.
.
.
> SET QUEUE/ENTRY=210/HOLD SYS$BATCH
```

The Submit command requests that the job MYJOB be placed in the batch queue. Before the job starts execution, the Set Queue command requests that the job be placed in a hold status. The job remains in that status until a subsequent Set Queue command releases it.

Notes:

- RSX-11M does not support the Set Queue command.
- See Section 2.7.6 for a description of absolute time.

SET RMS_DEFAULT

4.26.9 Set RMS Default

The Set RMS Default command defines default values for the multiblock and multibuffer fields used by VAX-11 RMS during file operations; it has no effect on RMS-11 file operations. You can set defaults for sequential or relative files on a process-only or system-wide basis.

Format:

```
SET RMS_DEFAULT [keywords]
```

```
keywords      /BLOCK_COUNT=count
               /BUFFER_COUNT=count
               /DISK
               /INDEXED
               /MAGTAPE
               /PROCESS
               /RELATIVE
               /SEQUENTIAL
               /SYSTEM
               /UNIT_RECORD
```

Examples:

```
> SET RMS_DEFAULT/BUFFER_COUNT=2
```

This command defines the current process's default multibuffer count for input/output operations as two. The default applies to all sequential file operations, that is, operations on sequential disk files, magnetic tapes, and unit record devices.

```
> SET RMS_DEFAULT/BLOCK_COUNT=4
```

This command defines the default multiblock count for this process as four. The default is not applied to user programs that establish explicit multiblock counts.

```
> SET RMS_DEFAULT /RELATIVE
```

This command indicates that, for the current process's disk files, the default file organization is relative. When you specify /RELATIVE or /SEQUENTIAL, /DISK is assumed.

SET TERMINAL

4.26.10 Set Terminal

The Set Terminal command changes the characteristics of a specified terminal for a particular application or overrides system default characteristics. The default characteristics are defined at system generation based on the most common type of terminal in use.

Format:

```
SET TERMINAL[=device-name] [keywords]
```

```
keywords      /[NO] BROADCAST
               /[NO] CRFILL [=formula]
               /[NO] ECHO
               /[NO] EIGHTBIT
               /[NO] ESCAPE
               /[NO] HARDCOPY
               /[NO] HOLDSCREEN
               /[NO] HOSTSYNC
               /[NO] INTERACTIVE
               /LA36
               /[NO] LFFILL [=formula]
               /[NO] LOCAL
               /[NO] LOWERCASE
               /PAGE=n
               /[NO] PARITY [=option]
               /[NO] PASSALL
               /[NO] READSYNC
               /[NO] REMOTE
               /[NO] SCOPE
               /SPEED=rate
               /[NO] TAB
               /[NO] TTSYNC
               /[NO] TYPEAHEAD
               /UNKNOWN
               /[NO] UPPERCASE
               /VT05
               /VT52
               /VT55
               /WIDTH=n
               /[NO] WRAP
```

Table 4-2 shows terminal defaults.

MCR COMMANDS

Table 4-2
Default Characteristics for Terminals

Terminal Characteristic	Terminal Type and Qualifier			
	UNKNOWN /UNKNOWN	LA36 /LA36	VT05 /VT05	VT5x /VT52 /VT55
WIDTH	*	132	72	80
PAGE	*	16	16	16
SPEED	*	*	*	*
BROADCAST	*	*	*	*
CRFILL	*	no	no	no
ECHO	*	yes	yes	yes
EIGHTBIT	*	no	no	no
ESCAPE	*	*	no	*
HOLDSCREEN	*	no	no	*
HOSTSYNC	*	no	no	yes
LFFILL	*	no	4	no
LOWERCASE	*	yes	no	yes
PASSALL	*	no	no	no
REMOTE	*	*	*	*
TYPEAHEAD	*	yes	yes	yes
TAB	*	no	no	yes
TTSYNC	*	yes	yes	yes
WRAP	*	yes	yes	yes

* The current setting is not changed by the qualifier.

Examples:

> SET TERMINAL /VT05

This command establishes the current terminal as a VT05, and sets its characteristics to that of a VT05.

> SET TERMINAL=TTA3 /NOTYPEAHEAD

This command establishes the terminal named TTA3 as one from which input is accepted only when a program issues a read to it; that is, the terminal's use is dedicated to an application program.

SET /UIC

4.26.11 Set Default UIC

The Set Default UIC command establishes a new directory as the default and, if you have Change Mode to Kernel privilege, establishes a new user identification code as the default.

Format:

SET /UIC=[g,m]

g	Specifies the UIC group number in the range of 0 through 377 (octal).
m	Specifies the UIC member number in the range of 0 through 377 (octal).

Examples:

> SET /UIC=[200,220]

This command establishes the UIC [200,200] as the default for the issuing process.

Notes:

- RSX-11M supports the Set Default UIC command.
- To display your UIC, use the Show Process command.

SET VERIFY

4.26.12 Set Verify

The Set Verify command controls whether lines in indirect command files are displayed at the terminal or printed in a batch job log when the indirect command file is executed. With verification off, only MCR commands are displayed or printed. With verification on, the entire command stream is displayed or printed. String substitution occurs before lines are displayed or printed.

Format:

SET [NO]VERIFY

Example:

```
> SET VERIFY
> @TEST.CMD
.
.
.
> SET NOVERIFY
```

In the sequence above, verification is turned on to test a new indirect command file. Each line in the file is displayed followed by related error messages, if any. After TEST.CMD terminates, the Set No Verify command restores the system default.

Notes:

- By default, the system does not display the command lines at the terminal when it processes indirect command files that are requested interactively, rather than queued for batch processing. System responses and error messages are always displayed.
- The default setting for batch jobs is for all lines in the indirect command file to appear in the batch job listing.
- If any lines displayed as a result of Set Verify contain symbol names that are substituted before command execution, the command interpreter displays the line as it appears after substitution.
- When you change the verification setting, it remains in effect for all indirect command files that you subsequently execute.

SET WORKING__SET

4.26.13 Set Working Set

The Set Working Set command redefines the default working set size for the process or sets an upper limit to which the working set size can be changed by an image that the process executes.

Format:

```
SET WORKING_SET [keywords]
```

```
keywords      /LIMIT=n
               /QUOTA=n
```

Example:

```
> SHOW WORKING_SET
Working Set /Limit=100 /Quota= 200 Authorized Quota= 200
> SET WORKING_SET/LIMIT=90
```

In the sequence above, the Show Working Set command displays the current working set limit, quota, and authorized quota for the process. The Set Working Set command reduces the limit to 90.

Notes:

- RSX-11M does not support the Set Working Set command.
- An RSX-11M image cannot affect the working set limit.

MCR COMMANDS

4.27 SHOW

The Show command provides a number of options that let you display characteristics and defaults for your process, jobs, and devices. Table 4-3 lists the Show options.

The Show command options are equivalent to the options of the DCL SHOW command. These options are detailed in the VAX/VMS Command Language User's Guide and summarized in the following sections.

Table 4-3
SHOW Command Options

Option	Displays
[DAY]TIME	The current date and time
DEFAULT	The current default disk and directory device
DEVICES	The status of devices in the system
LOGICAL	Current logical name assignments
MAGTAPE	Information about a designated magnetic tape unit
NETWORK	The availability of network nodes, including the current node
PRINTER	Information about a designated magnetic tape unit
PROCESS	Attributes of the current process, including privileges, resource quotas, memory usage, priority, and accounting information
PROTECTION	The current default protection applied to files
QUEUE	Printer or batch jobs that have been queued but not yet printed or processed
RMS_DEFAULT	The current default multiblock and multibuffer counts used by VAX-11 RMS for file operations
STATUS	The status of the current job, including accumulated CPU time, open file count, and count of I/O operations
SYMBOL	Current symbol definitions
SYSTEM	A list of all processes in the system
TERMINAL	The device characteristics of your terminal
TRANSLATION	The result of translating a logical name
WORKING_SET	The current working set size limit and quota

SHOW DAYTIME

4.27.1 Show Daytime

The Show Daytime command displays the current date and time of day.

Format:

SHOW [DAY]TIME

Example:

```
> SHOW TIME
31-DEC-1977 00:03:45
```

Note:

- RSX-11M does not support the Show Daytime command; it does, however, support the Time command, which VAX/VMS also supports.

SHOW DEFAULT

4.27.2 Show Default

The Show Default command displays the current default disk device and directory name.

Format:

SHOW DEFAULT

Example:

```
> SHOW DEFAULT  
DBA1:[122,020]
```

```
> SHOW DEFAULT  
DBB2:[CALEB]
```

Note:

- The default disk and directory are established in the user authorization file entry, but you can use the Set Default command to change them for the duration of a terminal session.

SHOW DEVICES

4.27.3 Show Devices

The Show Devices command displays the status of all devices or a set of devices in the system or the status of a particular device, or lists the devices that currently have volumes mounted on them and/or are allocated to processes.

Format:

SHOW DEVICES [keywords][device-name]

keywords /ALLOCATED
 /BRIEF
 /FULL
 /MOUNTED

Examples:

```
> SHOW DEVICES/MOUNTED
List of Devices
Device  Device      Device  on   3-JAN-1978  08:40:41.77
Name    Status   Characteristics  Count  Volume      Free  Trans  Mount
DBA0    on line  MNT              0      LABEL      24712  1      1
DBA3    on line  MNT              0      SYSSCR      7583  27     1
.
.
.
```

This command causes the names of all mounted devices to be displayed with additional information about each device and the volume it contains.

```
> SHOW DEVICES DB:
List of Devices
Device  Device      Device  on   3-JAN-1978  08:40:41.77
Name    Status   Characteristics  Count  Volume      Free  Trans  Mount
DBA0    on line  MNT              0      LABEL      24712  1      1
DBA3    on line  MNT              0      SYSSCR      7583  27     1
.
.
.
```

This command causes the names of all RP04, RP05, and RP06 disk drives to be displayed.

SHOW LOGICAL

4.27.4 Show Logical

The Show Logical command displays all logical names in one or more logical name tables, or displays the current equivalence name assigned to a specified logical name. By default, the Show Logical command displays all logical names in all logical name tables in the following format.

logical-name = equivalence name (table)

Format:

SHOW LOGICAL [keywords] [logical-name]

keywords	/ALL
	/GROUP
	/PROCESS
	/SYSTEM

Examples:

```
> SHOW LOGICAL MYDISK
MYDISK = DBB1: (Process)
```

This command causes the equivalence name for the logical name MYDISK and the logical name table in which it is found to be displayed.

```
> SHOW LOGICAL/PROCESS
```

Contents of process logical name table:

```
SYSS$INPUT = TTE7:
SYSS$OUTPUT = TTE7:
SYSS$ERROR = TTE7:
TT = TTE7:
SYSS$COMMAND = TTE7:
SYSS$DISK = DBA1:
MYDISK = DBB1:
```

This command causes all the entries in the process logical name table to be displayed with their equivalence names.

Note:

- The Show Logical command performs translation recursively up to 10 times.

SHOW MAGTAPE

4.27.5 Show Magtape

The Show Magtape command displays information about a designated magnetic tape unit.

Format:

SHOW MAGTAPE device-name

device name Specifies the magnetic tape unit. If you omit the controller or unit number, the command displays information for controller A and unit 0, respectively, by default.

Example:

```
> SHOW MAGTAPE MTB2
MTB2: UNKNOWN, DENSITY=800, FORMAT=Normal-11
End-of-File odd Parity
```

SHOW NETWORK

4.27.6 Show Network

The Show Network command displays the availability of the local node as a member of the network, and the names of all nodes that are currently accessible by the local node. If the network ACP is running, the Show Network command displays the local node name, number, and state (ON or SHUT). Then, it displays the names of accessible remote nodes and lines on which the nodes are accessed.

Format:

```
SHOW NETWORK
```

Example:

```
> SHOW NETWORK
NETWORK STATUS AS OF 06-JUN-1978 12:42
```

```
LOCAL NODE NAME: VAX1
NUMBER: 3
STATUS: ON
```

```
ACCESSIBLE REMOTE NODES
NODE LINE
MANILA: XMA0
CHI: XMC0
```

The Show Network command displays the status of the local node and lists available remote nodes.

SHOW PRINTER

4.27.7 Show Printer

The Show Printer command displays the default characteristics currently defined for a system printer.

Format:

SHOW PRINTER device-name

Example:

```
> SHOW PRINTER LPA0:  
LPA0:  LP11,  WIDTH=132,  PAGE=64,  NOCR,  FF,  LOWER
```

This command requests a display of the characteristics of printer LPA0.

SHOW PROCESS**4.27.8 Show Process**

The Show Process command displays information about the current process.

Format:

SHOW PROCESS [keywords]

keywords /ACCOUNTING
 /ALL
 /PRIVILEGES
 /QUOTAS
 /SUBPROCESSES

Example:

> SHOW PROCESS

```

3-JAN-1978 08:44:41.67      TTE7:      User: FIELDING
Pid : 0003002F      Proc. name : FIELDING      UIC : [122,20]
Priority : 5 Default file spec. :DBB2:[122,020]

Devices allocated : TTE7

```

SHOW PROTECTION

4.27.9 Show Protection

The Show Protection command displays the current default file protection that is applied to all files created during the terminal session; that is, it indicates which combination of the privileges read (R), write (W), execute (E), and delete (D) is to be extended to system, owner, group, and world.

Format:

SHOW PROTECTION

Example:

```
> SHOW PROTECTION  
SYSTEM=RWED, OWNER=RWED, GROUP=RWED, WORLD=R
```

This command indicates that the system, the file owner, and all others in the same group as the file owner have read, write, execute, and delete access; all other users have only read access.

SHOW QUEUE

4.27.10 Show Queue

The Show Queue command displays the current status of entries in the printer or batch job queues. By default, the Show Queue command provides a brief listing of information about jobs in the specific queue. It lists the job currently being processed and all your jobs that are in the queue.

Format:

```
SHOW QUEUE [keywords][queue-name]
```

```
keywords      /ALL
               /BATCH
               /BRIEF
               /DEVICE
               /FULL
```

Examples:

```
> SHOW QUEUE/DEVICE
```

```
* DEVICE QUEUE "LPA0:" FORMS=0 GENPRT FLAG
```

```
* DEVICE QUEUE "LPB0:" FORMS=0 GENPRT FLAG
```

```
Current job 138 CRAMER      ALPHA      , Pri=4, 12-JAN-1978 14
Pending job 139 HIGGINS     DOCMASTE, Pri=4, 12-JAN-1978 15
Pending job 140 HIGGINS     DOCRUN3   , Pri=4, 12-JAN-1978 15
```

```
* DEVICE QUEUE "SYS$PRINT" FORMS=0 GENDEV FLAG
```

```
Holding job 105 HIGGINS     DOCRUN2   , Pri=4, COPIES=1, 12-JAN-1978
```

The Show Queue command displays the status of the printer queues. The first queue, LPA0, has no entries. The second queue, LPB0, is currently processing a job for the user CRAMER. Two jobs are pending for the user HIGGINS (who issued the command). The third queue, named SYS\$PRINT, consists of jobs that are being held.

```
> SHOW QUEUE SYS$BATCH/FULL
```

```
* BATCH QUEUE "SYS$BATCH" INIPRI=4, SWAP
```

```
Current job 201 WILSON      BATCH01   Pri=4, 12-JAN-1978 14:15
Current job 202 HARRIS      FORCLG    Pri=4, 12-JAN-1978 14:20
Current job 203 HIGGINS     ASSEMBLE  Pri=4, 12-JAN-1978 14:22
    DBAL:ASSEMBLE.COM;3
Pending job 205 HIGGINS     BACKUP      12-JAN-1978 14:30
    DBAL:BACKUP.COM;2
    DBAL:AVERAGE.COM;2
    DBAL:DOCMaster.COM;5
```

The Show Queue command requests a display of all jobs in the batch job queue. The /FULL qualifier requests, for jobs submitted by the user typing the command, the file specifications of files in the job.

SHOW RMS_DEFAULT

4.27.11 Show RMS Default

The Show RMS Default command displays the current multiblock and multibuffer counts for the process and the system. VAX-11 RMS uses these counts for multiblock and multibuffer file operations.

Format:

```
SHOW RMS_DEFAULT
```

Example:

```
> SHOW RMS_DEFAULT
```

	MULTI- BLOCK COUNT	:	Indexed	:	Relative	:	MULTI-BUFFER COUNTS			
		:		:		:	Disk	Magtape	Unit	Record
Process	0	:	0	:	0	:	0	0	0	0
System	4	:	0	:	0	:	0	0	0	0

SHOW STATUS

4.27.12 Show Status

The Show Status command displays the status of the current process.

Format:

SHOW STATUS

Example:

```
> SHOW STATUS
Status on      3-JAN-1978   08:47:51.09   Elapsed CPU: 000:00:12.08
Buff. I/O :    181        Cur. ws. :    296   Open Files :    0
Dir. I/O :     42        Phys. Mem :    47    Page faults :   185
```

Note:

- If you interrupt a running image by pressing CTRL/Y and issue a Show Status command, you can continue image execution using the Continue command. You cannot continue execution of an interrupted image after using the Show Process command.

SHOW SYMBOLS

4.27.13 Show Symbols

The Show Symbols command displays the value currently assigned to a specified symbol, or displays all the symbols and their values in either the local or global symbol tables.

Format:

```
SHOW SYMBOLS [keywords] [symbol-name]
```

```
keywords      /ALL
               /GLOBAL
               /LOCAL
```

Examples:

```
> SHOW SYMBOLS/GLOBAL $SEVERITY
$SEVERITY = 1
```

This command displays the value of the global symbol SEVERITY.

```
> SHOW SYMBOLS/LOCAL/ALL
A = 1
B = 0
MODE = INTERACTIVE
N1 = 42.
```

This command displays all the entries in the local symbol table.

Notes:

- You can define symbols using MCR .ASKx or .SETx directives and by equivalent DCL functions.
- Local symbols are available only at the indirect level at which they are defined. Global symbols are available at all levels.
- The .ENABLE GLOBAL directive is used to allow the definition of global symbols.

SHOW SYSTEM**4.27.14 Show System**

The Show System command displays information relating to the activity in the system.

Format:

SHOW SYSTEM

Example:

```
> SHOW SYSTEM
VAX/VMS Processes on 3-JAN-1978 12:26:00.50
Pid      Process Name  UIC      State  Pri  Dir.I/O    CPU      Page flts  Ph.Mem
00010000  NULL                000,000  COM     1      0  13:35:18.42      0      0
00010001  SWAPPER             000,000  HIB    17      0  00:04:37.98      0      0
.
.
00070031  YOURJOB             122,020  CUR     6      53  00:00:01.56    384     56
```

SHOW TERMINAL

4.27.15 Show Terminal

The Show Terminal command displays the characteristics of your terminal or of another designated terminal.

Format:

```
SHOW TERMINAL    [device-name]
```

Example:

```
> SHOW TERMINAL
TTE7: /VT52, WIDTH=80  PAGE=16, OWNER=SELF
      SPEED=(2400,2400), CRFILL=0, LFFILL=0,  NO PARITY
      INTERACTIVE,  ECHO, TYPEAHEAD, NOESCAPE, HOSTSYNC, TTSYNC
      LOWERCASE,  TAB,  WRAP,  SCOPE,  LOCAL, NOHOLDSCREEN,
      NOEIGHTBIT,  BROADCAST, NOREADSYNC
```

SHOW TRANSLATION

4.27.16 Show Translation

The Show Translation command searches the process, group, and system logical name tables, in that order, for a specified logical name and returns the equivalence name of the first match found. The translation is not recursive; it is recursive with the Show Logical command.

You can interrupt an executing image by pressing CTRL/Y, issue a Show Translation command for a logical name and then resume execution with a Continue command. You cannot issue a Show Logical command and then resume image execution in this way.

Format:

```
SHOW TRANSLATION    logical-name
```

Example:

```
> SHOW TRANSLATION DK0
DK0 = DBB2: (System)
```

This show command displays the VAX/VMS physical device name DBB2: as the equivalence name for logical name DK0.

SHOW WORKING__SET

4.27.17 Show Working Set

The Show Working Set command displays the working set quota and limit assigned to the current process.

Format:

SHOW WORKING_SET

Example:

```
> SHOW WORKING_SET
Working Set /Limit=100    /Quota=200    Authorized Quota=200
```

STOP**4.28 STOP**

The Stop command terminates execution of one of the following:

- A program or indirect command file that was interrupted by CTRL/C or CTRL/Y
- A program executing in a subprocess or detached process

Format:

STOP [keyword] [process-name]

keyword /IDENTIFICATION=pid

process-name Specifies the name of the process or subprocess containing the program to be terminated.

Command Keyword:

/IDENTIFICATION=pid Specifies the process identification of the process or subprocess containing the program to be terminated.

Examples:

> @WARHOL.CMD

.

.

^Y

> STOP

This sequence starts the execution of the indirect file WARHOL.CMD, interrupts its execution by pressing CTRL/Y, and terminates its execution by typing STOP. By default, the image in the current process is terminated.

> RUN /PROCESS=PRIOR LIMNER.TSK

.

.

^Y

> STOP PRIOR

This sequence starts the execution of the image LIMNER.TSK in a subprocess named PRIOR. Typing STOP causes the image to be terminated and the subprocess to be deleted.

MCR COMMANDS

Notes:

- The current process does not require privilege to stop one of its subprocesses. The current process must have group or world privilege to stop any other process.
- The Stop command causes an abnormal termination of the image currently executing; if the image has declared any exit handling routines, they are not given control.

When an image is interrupted by CTRL/Y or CTRL/C, and another image is run, the interrupted image is also terminated. However, in this case, exit handling routines are allowed to execute before the next image runs.

- When an indirect command file is stopped, the entire procedure (that is, all levels) is terminated and control returns to the command interpreter.
- Stopping the image executing in a subprocess or detached process causes the specified subprocess or detached process to be deleted.
- When you create a subprocess or detached process using the Run command, the command displays the process identification of the process created.
- The Show Process command displays the current status of subprocesses.

SUBMIT**4.29 SUBMIT**

The Submit command queues an MCR indirect command file or a DCL command procedure for execution as a batch job. Your default (not current) command interpreter determines whether you can submit an indirect command file or a command procedure as a batch job. If your default command interpreter is MCR, you can submit MCR indirect command files for batch processing; if it is DCL, you can submit DCL command procedures for batch processing. All output from the indirect command file is queued to the device assigned the logical name SYS\$PRINT at the time the job terminates.

Format:

SUBMIT [keywords] file-spec [keyword],...

keywords /AFTER=absolute-time
 /([NO])DELETE
 /([NO])HOLD
 /([NO])IDENTIFY
 /NAME=job-name
 /PARAMETERS=parameters
 /PRIORITY=n
 /QUEUE=queue-name

file-spec Specifies the name of a command procedure to be submitted for batch execution. If no file type is specified, the Submit command uses the default file type of COM. If you specify more than one file, use either a comma (,) or a plus (+) to separate file specifications. Files are processed in the order of specification.

Command Keywords:

/AFTER=absolute-time Requests that the job be processed after a specific time of day. Specify the time according to the rules for specifying absolute times.

If the specified time has already passed, the job is queued for immediate processing.

By default, jobs are submitted immediately.

/HOLD
 /NOHOLD (default) Controls whether the job is to be made available for immediate processing. If you specify /HOLD, the job is not released until you issue a Set Queue /NOHOLD command for it.

MCR COMMANDS

/IDENTIFY (default)
/NOIDENTIFY

Controls whether the command interpreter displays the job identification assigned to the job and the name of the device to which it has been queued, as follows.

JOB jobid ENTERED ON QUEUE device-name

Usually, the device-name is SYS\$BATCH.

/NAME=job-name

Defines a 1- to 8-character alphanumeric name string to identify the job. The name string is displayed in response to the Show Queue command.

If you do not specify a name, the name string defaults to the file name of the first file, truncated to eight characters, if necessary.

Output files from the indirect command file are named job-name.LOG.

/PARAMETERS=parameters

Specifies from one to eight optional parameters to be passed to the indirect file. The parameters define values that are equated to the symbols P1, P2, ... P8 in the indirect command file. The symbols are local to the specified indirect command file.

If you specify more than one parameter, separate them with commas and enclose them in parentheses.

The commas delimit the parameters. To specify a parameter that contains special characters or delimiters, enclose the parameter in quotation marks.

/PRIORITY=n

Specifies the priority for the job being submitted. The value of n must be in the range of 0 through 31; 0 is the lowest priority, and 31 is the highest.

By default, jobs are assigned the same priority as your process. You can assign lower priorities to nonessential jobs to allow your other jobs to be processed sooner.

Privilege is required to set a priority value that is higher than your process's priority.

/QUEUE=queue-name

Specifies the name of a particular batch job queue to which the job is to be submitted.

MCR COMMANDS

File Keyword:

<code>/DELETE</code> <code>/NODELETE</code> (default)	Indicates whether the associated indirect command file is to be deleted after processing. If you specify this keyword as a command keyword, all files specified are deleted after processing.
---	---

Examples:

```
SUBMIT SHEELER
JOB 134 ENTERED ON QUEUE "SYS$BATCH"
```

This command submits the indirect command file SHEELER.CMD for processing. The job is assigned the job identification 134 and placed in the default batch queue, SYS\$BATCH.

```
SUBMIT BURCHFIELD /PARAMETERS="DBA2:[PAINTER]NATURE.DAT"
JOB 123 ENTERED ON QUEUE "SYS$BATCH"
```

This command submits the indirect command file BURCHFIELD and defines the symbol P1 as being equal to the file specification DBA2:[PAINTER]NATURE.DAT.

Notes:

- When the indirect command file is dequeued for execution, it is processed by the default command interpreter of the user who submitted the job. VAX/VMS does not consider the command interpreter used to submit the job.
- The file or files queued for process using the Submit command are considered a job. The system assigns a unique job identification to each job. You can use the Show Queue command to determine the job identification of a queued batch job.
- The attributes of a queued job can be changed using the Set Queue command.
- You can use the Delete/Entry command to remove a job from a batch queue or to delete an entry that is being processed.
- RSX-11M does not support the Submit command.

TIME

4.30 TIME

The Time command displays the current date and time.

Format:

TIME

Example:

```
> TIM
22-FEB-1978 12:43:12
```

TYPE**4.31 TYPE**

The Type command displays the contents of a file or group of files on the terminal.

Format:

TYPE [/OUTPUT=output-file-spec] input-file-spec,...

/OUTPUT=output-file-spec

Requests that the output from the Type command be written to the specified file rather than displayed on your terminal.

By default, the Type command displays the requested files at the requesting terminal.

input-file-spec

Is the file specification of a file to be displayed.

If multiple files are specified, they can be separated with either a comma (,) or a plus sign (+). In either case, the files are displayed in the order named.

You can use a wild card instead of the directory, file name, file type, or version fields. The Type command displays all files that satisfy the file description in alphanumeric order for Files-11 Structure Level 2. Files are displayed in random order for Files-11 Structure Level 1.

If you omit the file type, the Type command uses LIS by default.

Example:

> TYPE POLLOCK

The Type command locates POLLOCK.LIS in the default directory and types it on the terminal.

Notes:

- To temporarily halt the output, use CTRL/S. To resume output after CTRL/S, use CTRL/Q.
- To suppress the display while allowing processing to continue, use CTRL/O. If CTRL/O is pressed again before the command terminates, output resumes at the current point in command processing.

If you have entered more than one file specification or used wild cards, CTRL/O suppresses the display of the current file only. Output resumes when the Type command begins the next file.

- To stop the display completely, press CTRL/Y and then use the Stop command.

UFD

4.32 UFD

The UFD command creates a user file directory or subdirectory on a Files-11 Structure Level 1 or 2 volume and enters the name of the UFD in the volume's master file directory. The volume must have been initialized and must be mounted before you can define UFDs for it.

To create a user file directory, you must have access to the master file directory. The protection and UIC for the master file directory are established during volume initialization; they are the volume protection and volume owner UIC.

Format:

```
UFD devcu:[directory] [/keywords]
```

keywords /ALLOC=number-entries
 /PRO=[system,owner,group,world]
 /UIC=[g,m]

devcu Specifies the device, controller, and unit number of
 the device containing the volume.

[directory] Specifies the directory or subdirectory name. If you
 create a subdirectory on a structure level 1 volume,
 RSX-11M cannot read it.

Command Keywords:

/ALLOC=number-entries Specifies the number of directory
 entries for which space is to be
 allocated. The number provided is
 rounded up to the next multiple of 32
 (decimal).

 If you do not specify /ALLOC, the
 command allocates space for 32 directory
 entries.

/PRO=[system,owner
 group,world] Establishes access rights for the
 directory file. Specify protection
 codes for each user category.

 Read (R), write (W), and delete (D)
 protection are identical when the volume
 is being accessed under either VAX/VMS
 or RSX-11M. The E field indicates
 execute under VAX/VMS and extend under
 RSX-11M.

 If you do not specify /PRO, the command
 supplies the default protection of the
 directory (MFD, UFD, or subdirectory) in
 which the new directory is being
 created.

 If you omit any one of the protection
 categories, users in that category are
 denied access to the directory.

MCR COMMANDS

/UIC=[g,m]

Specifies the owner UIC for the directory or subdirectory. If you do not specify /UIC, the UFD command by default uses the owner UIC of the directory in which the new directory is being created.

Example:

```
> MOU DBB2:MYVOL
> UFD DBB2:[230,222]/PRO=[RWED,RWED,RE,R]
```

The first command mounts the volume labeled MYVOL. The UFD command requests the creation of a directory with UIC [230,222] on the volume and establishes the protection for that directory.

Note:

- VAX/VMS does not support the /UF switch for PIP to create directories.

CHAPTER 5

INDIRECT COMMAND FILES

VAX/VMS supports RSX-11M indirect command files to provide the same functions that RSX-11M provides:

- Indirect command files for components, for example, MACRO-11 and the task builder
- MCR indirect command files

Indirect command files intended for processing by RSX-11M components function identically under VAX/VMS and RSX-11M, as described in Chapter 3.

MCR indirect command files that are to execute under VAX/VMS can contain the following:

- Any of the indirect command file processor directives summarized alphabetically in Table 5-1
- Any of the VAX/VMS MCR commands described in Chapter 4

MCR indirect command files can be nested; refer to Section 5.14.

Unlike RSX-11M, which provides an indirect command file processor task (AT.) to interpret directives, the VAX/VMS MCR command interpreter processes both MCR commands and directives. As a result, you can type directive sequences at the terminal and test them without creating and executing an indirect command file.

VAX/VMS support of indirect command files is intended to allow existing files to execute under VAX/VMS. Because it is intended for existing files, VAX/VMS does not perform such stringent syntax checking within the indirect command file as RSX-11M does. As a result, indirect command files developed under VAX/VMS may not execute successfully under RSX-11M.

Table 5-1
Summary of VAX/VMS Indirect Command File Processor Directives

Format	Function	Differences between VAX/VMS and RSX-11M
/	Close all data and command files and exit	None
.label:	Define a label	None
!text or ;text	Provide commentary	None
.ASK sssss txt-string	Ask a question and wait for a reply	None
.ASKN sssss txt-string	Ask for definition of a numeric symbol	Can specify hexadecimal values; see Section 5.7.
.ASKN[low:high] sssss txt-string ¹	Ask for definition of a numeric symbol specifying an allowable range	Can specify hexadecimal values; see Section 5.7.
.ASKN[:def] sssss txt-string ¹	Ask for definition of a numeric symbol specifying the default value	Can specify hexadecimal values; see Section 5.7.
.ASKN[low:high:def] sssss txt-string ¹	Ask for definition of a numeric symbol specifying an allowable range and the default value	Can specify hexadecimal value; see Section 5.7.
.ASKS sssss txt-string	Ask for definition of a string symbol	None
.ASKS[low:high] sssss txt-string ¹	Ask for definition of a string symbol specifying the allowable number of characters	None
.CHAIN file-spec	Continue processing using another file	None

¹ Brackets are required syntax in these directives only. Brackets in other directives indicate optional features.

(continued on next page)

Table 5-1 (Cont.)
Summary of VAX/VMS Indirect Command File Processor Directives

Format	Function	Differences between VAX/VMS and RSX-11M
.CLOSE [#n]	Close secondary file	None
.DATA [#n] txt-string	Write data to secondary file	None
.DEC ssssss	Decrease value of numeric symbol by 1	None
.DELAY nnu	Delay indirect command file execution for the specified period of time	VAX-11/780 hardware provides a 100 ticks-per-second clock; under VAX/VMS, hours cannot be specified.
.DISABLE DATA [#n]	Disable writing of data to secondary file	None
.DISABLE DOLLARS	Disable dollar sign recognition	See Section 5.11. RSX-11M does not support this directive.
.DISABLE ESCAPE	Disable use of escape character as response to .ASK, .ASKN, or .ASKS	Ignored by VAX/VMS. See Section 5.3.
.DISABLE GLOBAL	Disable definition of global symbols	None
.DISABLE LOWERCASE	Enable lowercase to uppercase conversion	None
.DISABLE SUBSTITUTION	Disable symbol substitution	None
.ENABLE DATA [#n]	Enable writing of data to a secondary file	None
.ENABLE DOLLARS	Enable dollar sign recognition	See Section 5.11. RSX-11M does not support this directive.

(continued on next page)

Table 5-1 (Cont.)
Summary of VAX/VMS Indirect Command File Processor Directives

Format	Function	Differences between VAX/VMS and RSX-11M
.ENABLE ESCAPE	Enable use of escape character as response to .ASK, .ASKN, or .ASKS	Ignored by VAX/VMS. See Section 5.3.
.ENABLE GLOBAL	Enable definition of global symbols (\$ssss)	None
.ENABLE LOWERCASE	Enable use of lower case characters in strings	Embedded multiple tabs and spaces are not compressed.
.ENABLE SUBSTITUTION	Enable symbol substitution	None
.GOSUB label	Call a subroutine	None
.GOTO label	Branch to a label	None
.IF ssssss relop expr	Determine whether symbols meet specified conditions	None
.IF[N]ACT ttttt	Determine whether a task is active	Response always indicates that the task is inactive; see Section 5.2.
.IF[N]DF ssssss	Determine whether a symbol is defined	None
.IFF ssssss	Determine whether a symbol is false	None
.IF[N]INS ttttt	Determine whether a task is installed	Response always indicates that the task is already installed; see Section 5.2.

(continued on next page)

Table 5-1 (Cont.)
Summary of VAX/VMS Indirect Command File Processor Directives

Format	Function	Differences between VAX/VMS and RSX-11M
.IF [N]LOA dd:	Determine whether a loadable driver is loaded	Response always indicates that the driver is loaded; see Section 5.2.
.IFT ssssss	Determine whether a symbol is true	None
.INC ssssss	Increase the value of a numeric symbol by 1.	None
.OPEN [#n] file-spec	Open a secondary file for output	None
.PAUSE	Pause for user action	Use the Continue command to resume indirect command file processing.
.RETURN	Return from subroutine	None
.SETF ssssss	Set the value of a symbol to false	None
.SETN ssssss numexp	Set a symbol to a numeric value	Can specify hexadecimal value; see Section 5.7
.SETS ssssss strexp	Set a symbol to a string value	None
.SETT ssssss	Set the value of a symbol to true	None
.TEST ssssss	Test the length of a string symbol	None

INDIRECT COMMAND FILES

5.1 REQUESTING INDIRECT COMMAND FILE EXECUTION

Under VAX/VMS, you request execution of an indirect command file in the same way that you do under RSX-11M; that is, by preceding the file specification of the indirect command file with an at sign (@). However, under VAX/VMS, additional options are available:

- You can use the /OUTPUT keyword to request that all output to SYS\$OUTPUT be written to the specified file or device.
- You can pass parameter values (P1 through P8) to the indirect file.

Section 4.15 contains a description of these options.

5.2 SUPPORT OF .IFINS, .IFACT, AND .IFLOA

The following directives test conditions that are not defined in the VAX/VMS environment:

- .IFINS and .IFNINS
- .IFACT and .IFNACT
- .IFLOA and .IFNLOA

VAX/VMS supports these directives by always making the following standard responses to them:

- Task is installed; that is, .IFINS is true and .IFNINS is false.
- Task is inactive; that is, .IFACT is false and .IFNACT is true.
- Driver is loaded; that is, .IFLOA is true and .IFNLOA is false.

Under VAX/VMS, users do not load drivers. The system manager determines which drivers are present.

5.3 SUPPORT OF .ENABLE AND .DISABLE ESCAPE

VAX/VMS ignores any .ENABLE ESCAPE or .DISABLE ESCAPE directives encountered in an indirect command file. Both the RETURN and ESCAPE keys always are recognized as valid line terminators. Pressing the ESCAPE key causes the symbol <ESCAPE> to be set.

5.4 UNSUPPORTED DIRECTIVES

VAX/VMS does not support the following indirect command file processor directives:

- .XQT and .WAIT
- .ONERR

INDIRECT COMMAND FILES

All other directives are supported. The following sections describe differences in directive support.

5.4.1 .XQT and .WAIT Alternative

As an alternative to .XQT and .WAIT, you can use the Run command. Because images execute serially in your process, the .WAIT directive is not needed; one image must terminate before the next begins.

If you use the Run command to create a subprocess or detached process, there is no means of synchronizing image execution from the indirect command file.

5.4.2 .ONERR Alternative

The VAX/VMS On statement, described in Section 4.22.2, can serve as an alternative to the .ONERR directive. However, the On statement provides a more general error handling capability than .ONERR. The MCR command interpreter does not restrict use of error handling to the specific errors associated with the .ONERR directive.

5.5 SWITCHES

RSX-11M defines three switches for use with indirect file specifications:

- /TR (trace) and /NOTR
- /DE (delete) and /NODE
- /MC (pass commands to MCR) and /NOMC

The defaults under RSX-11M are /NOTR, /NODE, and /MC. That is, indirect command files are not traced during execution and not deleted after processing, and the MCR command interpreter processes all commands.

VAX/VMS supports the default case and also allows use of the /DE switch to delete files after processing. It does not support the /TR and /MC switches. No deletion (/DE or /NODE) is the default.

As an alternative to /TR, you can use the Set Verify command to display command lines as they are processed; see Section 4.26.12.

5.6 SPECIAL SYMBOLS

VAX/VMS supports all the special symbols defined by RSX-11M. Table 5-2 summarizes the symbols and provides VAX/VMS-specific information where appropriate.

INDIRECT COMMAND FILES

Table 5-2
Special Symbols

Symbol	Meaning	VAX/VMS-Specific Information
<ESCAPE>	True if last query was answered with a single escape character	--
<DEFAULT>	True if answer to last .ASKN was defaulted	--
<ALPHAN>	True if last string entered as answer to .ASKS or tested with .TEST contains only alphanumeric characters	--
<RAD50>	True if last string entered as answer to .ASKS or tested with .TEST contains only Radix-50 characters	--
<MAPPED>	True if the system on which the image is running is mapped and false if it is unmapped	This symbol is always true.
<MEMSIZ>	Memory size in K words	Memory size is always 32K.
<SYUNIT>	Unit number of system device	Always 0
<STRLEN>	Length of last string entered as answer to .ASKS or tested by .TEST	
<SYDISK>	Device mnemonic of system device	The mnemonic always is SY.
<UIC>	Current directory	The current directory is in a valid VAX/VMS format.
<SYSUIC>	System UIC	Always [1,54]

Symbol substitution for special symbols and user-defined symbols is the same under both RSX-11M and VAX/VMS MCR.

5.7 NUMERIC SYMBOLS

Under VAX/VMS a numeric symbol defined using .SETN or .ASKN can be in one of the following radices:

- Octal (default radix)
- Decimal
- Hexadecimal

INDIRECT COMMAND FILES

You can specify the radix of a numeric value in either of the following ways:

- Using RSX-11M conventions, that is, no radix indicator or a leading pound sign (#) for octal, and a trailing period for decimal
- Using a VAX/VMS radix indicator

VAX/VMS radix indicators have the format %rn. The percent sign is required syntax; r indicates the radix (O for octal, D for decimal, and X for hexadecimal); and n is the value of the symbol. Regardless of the method used to specify the radix, the system stores and displays the value as a numeric string with the radix implied. The following are examples.

```
> .SETN A 22
> .SETN B %X22
> .SETN C 22.
> .SETN D %O07
> SHOW SYMBOLS/ALL
A = 22
B = 22
C = 22
D = 7
```

You cannot determine the radix of a symbol value using the Show Symbols command.

5.8 LABEL TABLE ENTRIES

As in RSX-11M, VAX/VMS places any label that appears on a line by itself in a label table so that, when a label is referred to, the command interpreter can locate it quickly. Labels defined on a line that also contains a command or other text are not placed in the label table but can be specified as the destination in .GOTO and .GOSUB directives.

The VAX/VMS MCR command interpreter establishes a new label table for each level of indirect command file. When a nested level exits, the current table is emptied to free space for the next higher level. Therefore, labels defined in an indirect command file must be unique within that file.

Labels passed over as a result of a .GOTO or .GOSUB directive are not placed in the label table. The label table contains only labels encountered in the actual execution stream.

5.9 SYS\$INPUT AND SYS\$COMMAND

VAX/VMS maintains two process-permanent files for command input: SYS\$INPUT and SYS\$COMMAND. SYS\$INPUT is the prime source of command input. That is, for an interactive user, SYS\$INPUT is the user's terminal; for indirect command file execution, SYS\$INPUT is the file of commands and data. SYS\$COMMAND is a secondary input source during execution of an indirect command file. It remains assigned to the initiating terminal. You can interrupt the processing of an indirect command file, issue a set of commands by means of SYS\$COMMAND, and continue the indirect command file without altering its SYS\$INPUT.

INDIRECT COMMAND FILES

All of the MCR commands described in Chapter 4 have separate streams for SYS\$INPUT and SYS\$OUTPUT. However, to be compatible with RSX-11M, RSX-11M commands such as PIP, EDI, TKB, and any other commands invoked by typing their image file name have SYS\$INPUT assigned to SYS\$COMMAND. That is, attempts to read from TI (SYS\$INPUT) actually result in reading from SYS\$COMMAND. This assignment allows the indirect command file to obtain information (for example, edits) from the terminal, as is often done under RSX-11M.

For cases in which the executing image is to read from SYS\$INPUT (that is, when it expects to read records from the indirect command file), you can use the Run command to request the image in the indirect command file. For example:

```
RUN SYS$SYSTEM:PIP
```

5.10 .ASKN DIRECTIVE

When VAX/VMS executes an .ASKN directive, it displays the default radix of the value after the text string that requests the value. The default radix is displayed as [O] (octal), [D] (decimal), or [H] (hexadecimal). Similarly, the response can be octal, decimal, or hexadecimal. To override the default, you can follow the RSX-11M conventions (a leading pound sign forces octal and a trailing period forces decimal), or you can specify a VAX/VMS radix operator in the format %rn. No operator is required to specify a value in the default radix.

The response to the request for a symbol value can be a symbol name or an expression. If hexadecimal is the default radix and you want to enter a hexadecimal value that starts with A through F, you must enter a leading zero or use the radix operator %X to distinguish the hexadecimal value from a symbol name.

5.11 .ENABLE DOLLARS DIRECTIVE

Under RSX-11M, MCR indirect command files can contain only commands and directives. Under VAX/VMS, MCR indirect command files can contain commands, directives, and data. For example, an indirect command file can contain the command to run a program followed by the data read by that program.

The .ENABLE DOLLARS directive indicates to the VAX/VMS MCR command interpreter that the indirect command file is to take advantage of the command interpreter's capability to distinguish commands and directives from data. The command interpreter distinguishes between the two by checking for dollar signs (\$). Once dollar sign recognition is enabled, you must precede all subsequent command and directive lines with a dollar sign, for example, \$RUN.

The MCR command interpreter also uses the dollar sign convention to prevent user programs from reading past the end of data in the indirect command file and to bypass any data records that remain unread after the user program terminates. That is, if a user program does not read all its data, the command interpreter can throw out the remaining data records and start processing the next command or directive.

INDIRECT COMMAND FILES

The following is an example of a VAX/VMS MCR indirect command file containing data.

```
.ENABLE DOLLARS
$TIM
$RUN MYPROG
data to be read by MYPROG
.
.
.
$TIM
.DISABLE DOLLARS
```

By default, dollar sign recognition is disabled at the beginning of an indirect command file level.

5.12 .PAUSE DIRECTIVE

The .PAUSE directive functions the same way under VAX/VMS as it does under RSX-11M. That is, once the pause starts, you can run other images and utilities in your process. While in a pause condition, the process's SYS\$INPUT is equated to SYS\$COMMAND so that any images run during the pause can read input from the terminal. To continue from the pause, type the Continue command.

If you interrupt image execution using CTRL/Y or CTRL/C during a pause and then issue a Continue command, the pause is terminated. When the image exits, the indirect command file resumes. If you issue a Stop command to terminate the image, the command interpreter closes the indirect command file and returns to interactive command level in addition to terminating the image.

5.13 .DELAY DIRECTIVE

Under VAX/VMS, you can use the .DELAY directive to delay indirect command file processing for a number of ticks, seconds, or minutes; you cannot specify a delay of a number of hours. To wait for an hour or more, specify an equivalent number of minutes, as follows.

```
.DELAY 90M
```

This directive above causes a delay of 1 hour and 30 minutes.

Pressing CTRL/Y interrupts a delay. If you type a Continue command, indirect command file processing resumes. A Stop command terminates indirect command file processing.

5.14 NESTING MCR INDIRECT COMMAND FILES

The VAX/VMS MCR command interpreter allows the nesting of four indirect command files each having eight subroutines. As the number of subroutines decreases, the number of nested files allowed increases, and vice versa.

The number of nested indirect command files allowed by RSX-11M components running under VAX/VMS (for example, TKB) is controlled by individual components, and is the same as under RSX-11M.

5.15 LEXICAL FUNCTIONS

VAX/VMS provides a set of lexical functions that return information about character strings and attributes of the current process. You can use lexical functions in MCR directives that accept expressions as parameters, for example, .SETN and .SETS. Table 5-3 summarizes the lexical functions.

Table 5-3
Summary of Lexical Functions

Function	Value Returned
'F\$DIRECTORY()	Current default directory name
'F\$EXTRACT(position,length,string)	Substring beginning in specified position for length specified of indicated string
'F\$LENGTH(string)	Length of specified string
'F\$LOCATE(substring,string)	Starting position of specified substring within string indicated; or, the length of the string if the substring is not found
'F\$LOGICAL(logical-name)	Equivalence name of specified logical name (first match found in ordered search of process, group, and system logical name tables); or, logical name if no match is found
'F\$MESSAGE(message-code)	Message string associated with a system status value
'F\$MODE()	INTERACTIVE or BATCH
'F\$PROCESS()	Current process name
'F\$TIME()	Current date and time of day
'F\$USER()	Current user identification code (UIC)
'F\$VERIFY()	TRUE (that is, a numeric value of 1) if verification is set on; FALSE (that is, a numeric value of 0) if verification is set off

The VAX/VMS Command Language User's Guide provides a more detailed explanation of lexical functions; the following paragraphs summarize syntax rules.

INDIRECT COMMAND FILES

The general format of a lexical function is:

`'F$function=name([args,...])`

<code>'F</code>	Indicates that a function name follows. The substitution operator (<code>'</code>) is required.
<code>function-name</code>	Specifies the function to be returned. All function names are keywords. You can truncate function names to any unique truncation.
<code>()</code>	Encloses function arguments, if any. The parentheses are required for all functions including those that do not accept arguments.
<code>args,...</code>	Specify the arguments for the function. You can specify arguments using symbol names, numeric literals, or string literals enclosed in quotation marks. Functions cannot be specified as arguments for functions. Literal strings specified as arguments cannot contain embedded symbols.

Section 5.17.2 contains an example of the use of lexical functions.

In the MCR command language, lexical functions are intended for use with the `.SETx` directives. If they are used in `.IF` directives, results may not be as expected. To make full use of lexical functions, you should use DCL rather than MCR.

5.16 OPERATIONAL NOTES

The notes below describe operational aspects of indirect command file processing under VAX/VMS.

- As in RSX-11M, all data files are closed when an indirect command file terminates and returns to the interactive command level, that is, when MCR prompts for the next command.
- When a command file terminates and returns to the interactive command level, the command interpreter clears the global symbol table.
- When lowercase is enabled, responses to `.ASKS` directives are treated as quoted strings; that is, the command interpreter does not change what is typed. As a result, embedded multiple tabs or spaces are not compressed.
- Under VAX/VMS, you cannot press CTRL/Z in response to a request for input (for example, `.ASK`) to terminate the indirect command file. You can, however, type CTRL/Y or CTRL/C to terminate the indirect command file at any point during processing.

5.17 RSX-11M AND RSX-11S SYSTEM GENERATION

VAX/VMS uses the MCR indirect command file capability to serve as the host system for RSX-11M and RSX-11S system generation. The actual system generation proceeds as it does under RSX-11M; however, a few steps must be taken in preparation for a system generation:

- You must have the Change Mode to Kernel privilege, or the target disk volume must be unprotected. Change Mode to Kernel privilege is required to perform a SET/UIC command that changes both the default directory and UIC. Section 5.17.1 describes the creation of an unprotected disk volume.
- Because the VAX/VMS version of the RSX-11M task builder produces image files with a file type of EXE, you must use RSX-11M's own version of its task builder from a mapped system to obtain files with a type of TSK. It must be in directory [1,54].
- You must assign equivalence names for RSX-11M device names contained in the system generation indirect command files and for TKB and SRD. Section 5.17.2 provides an indirect command file that can be run to make the necessary logical name assignments.

5.17.1 Creation of an Unprotected Disk

To create a totally unprotected disk volume, use the following steps:

- Initialize the disk volume specifying read, write, execute, and delete privileges for system, owner, group, and world.
- Create on that volume all of the directories normally created by UFD commands in the system generation indirect command file. Specify read and write access for all user categories in all directories.
- Change the protection of the master file directory to allow read/write access for all categories of user by issuing the following command.

```
SET PROTECTION [0,0]000000.DIR;1/PROTECTION=(S:RW,O:RW,G:RW,W:RW)
```

5.17.2 Pre-System Generation Indirect Command File

Figure 5-1 lists the command file required to make logical name assignments. This command file contains standard MCR commands and directives, and also takes advantage of VAX/VMS parameter passing capabilities and lexical functions as described in Sections 4.15 and 5.15, respectively.

INDIRECT COMMAND FILES

```
.ENABLE SUBSTITUTION
.IF P1 EQ "" .ASKS P1 TARGET DEVICE NAME ! Equates P1 to target device
! name typed by user

.SETS COLON ":"
.SETN L1 'F$LOCATE(COLON,P1)' ! Sets L1 equal to the
! location of the colon(:),
! if any, if string P1.
! Otherwise sets L1 equal to
! length of P1.

.IF L1 EQ 'F$LENGTH(P1)' .SETS P1 "'P1'"COLON'" ! If P1 does not
! contain a terminating
! colon, sets P1
! equal to target dev:

SHOW SYMBOL P1
SET DEFAULT 'P1' ! Sets default to target disk
ASN 'P1'=SY: ! Assigns SY, SY0, LB, LB0 to
! target disk
ASN 'P1'=SY0:
ASN 'P1'=LB:
ASN 'P1'=LB0:
ASN MP:=MP0: ! Equates MP0 and MP
ASN 'P1'[1,54]BIGTKB.TSK=TKB ! Assigns equivalence names
ASN 'P1'[1,54]SRD.TSK=SRD ! for TKB and SRD
```

Figure 5-1 Pre-System Generation Indirect Command File

Image files produced by the task builder referred to in this indirect command file have a file type of TSK rather than EXE.

APPENDIX A

PRIVILEGES

The following is a list of the privileges defined by VAX/VMS.

ACNT	Allow process to suppress accounting messages for created processes
ALLSPOOL	Allow process to allocate spooled devices
ALTPRI	Allow process to set priority values
BUGCHK	Allow process to make bug check error log entries
CMEEXEC	Allow process to change mode to executive
CMKRNL	Allow process to change mode to kernel and change UICs
DETACH	Allow process to create detached processes
DIAGNOSE	Allow process to issue diagnostic I/O requests
EXQUOTA	Allow process to exceed resource quota
GROUP	Allow process to control other processes in the same group
GRPNAM	Allow process to place names in the group logical name table
LOG_IO	Allow process to issue logical I/O requests to a device
MOUNT	Allow a process to execute a Queue I/O Request system service to mount a device
NETMBX	Allow process to create a network device
OPER	Allow process to have operator privileges
PHY_IO	Allow process to issue physical I/O requests to a device
PRMCEB	Allow process to create permanent common event flag clusters
PRMGBL	Allow process to create permanent global sections
PRMMBX	Allow process to create permanent mailboxes
PSWAPM	Allow process to alter its swap mode

PRIVILEGES

SETPRV	Allow process to set other process's privileges
SYSGBL	Allow process to create system global sections
SYSNAM	Allow process to place names in the system logical name table
TMPMBX	Allow process to create temporary mailboxes
VOLPRO	Allow process to override volume protection
WORLD	Allow process to control all other processes in the system

APPENDIX B
RESOURCE QUOTAS AND LIMITS

The following is a list of the resource quotas and limits defined by VAX/VMS.

ASTLM	AST limit quota
BIOLM	Buffered I/O limit
BYTLM	Buffer space quota
CPULM	CPU time limit
DIOLM	Direct I/O limit
FILLM	Open file quota
PGFLQUOTA	Paging file quota
PRCLM	Subprocess quota
TQELM	Timer queue entry quota
WSDEFAULT	Default working set size
WSQUOTA	Working set size quota

INDEX

A

Allocate Device command, 4-1
Allocating devices, 3-2
<ALPHAN>, 5-8
Append command, 4-3
.ASKN directive, 5-10
Assign command, 4-6

B

BAD (bad block utility), 1-3
Bye command, 4-8

C

Cancel command, 4-9
CL, 2-12
CO, 2-12
Command interpreter,
 selecting a, 3-2
Commands,
 Allocate Device, 4-1
 Append, 4-3
 Assign, 4-6
 Bye, 4-8
 Cancel, 4-9
 Commenting lines, 2-16
 Continuation of command
 lines, 2-15
 Continue, 4-11
 Copy, 4-12
 Deallocate, 4-16
 Debug, 4-17
 Delete, 4-18
 Deposit, 4-21
 Directory, 4-24
 Dismount, 4-27
 Examine, 4-29
 Execute procedure (@), 4-32
 Exit, 4-34
 Initialize, 4-36
 Keyword placement, 2-17
 Link, 4-43
 Login procedure, 4-49
 Logout, 4-50
 Mount, 4-51
 On Control Y, 4-60
 On severity-level, 4-62
 On statement, 4-59
 Print, 4-64
 Purge, 4-68
 Run, 4-69
 Set, 4-79
 Set Card Reader, 4-81

Commands (Cont.),

 Set Control Y, 4-82
 Set Default, 4-83
 Set Magtape, 4-84
 Set On, 4-85
 Set Process, 4-86
 Set Protection, 4-87
 Set Queue, 4-88
 Set RMS Default, 4-89
 Set Terminal, 4-90
 Set /UIC, 4-92
 Set Verify, 4-93
 Set Working Set, 4-95
 Show, 4-95
 Show Daytime, 4-96
 Show Default, 4-97
 Show Devices, 4-98
 Show Logical, 4-99
 Show Magtape, 4-100
 Show Network, 4-101
 Show Printer, 4-102
 Show Process, 4-103
 Show Protection, 4-104
 Show Queue, 4-105
 Show RMS Default, 4-106
 Show Status, 4-107
 Show Symbols, 4-108
 Show System, 4-109
 Show Terminal, 4-110
 Show Translation, 4-111
 Show Working Set, 4-112
 Stop, 4-113
 Submit, 4-115
 Summary of, 1-4 - 1-8
 Time, 4-118
 Type, 4-119
 UFD (user file directory),
 4-120
Continue command, 4-11
Control keys, 2-13 - 2-15
Controller designation, 2-7
Copy command, 4-12

D

Dates,
 specifications of, 2-18
Deallocate command, 4-16
Debug command, 4-17
<DEFAULT>, 5-8
Default file types, 2-9
Defaults in file specifications, 2-8
.DELAY directive, 5-11
Delete command, 4-18
Deposit command, 4-21

INDEX (Cont.)

Device mapping, 2-12, 2-13
Device names, 2-6
 generic, 3-3
Devices,
 allocation of, 3-2
Devices supported, 2-7
Directives,
 MCR indirect command file
 processor directive, 5-2
Directives (MCR),
 summary of, 5-2
 unsupported, 5-6
Directory command, 4-24
Directory names, 2-7
.DISABLE directive, 5-6
.DISABLE DOLLARS directive,
 5-10
Disk,
 creation of unprotected,
 5-14
Dismount command, 4-27
DMP (file dump utility), 1-3

E

EDI, 1-2
.ENABLE DOLLARS directive,
 5-10
.ENABLE directive, 5-6
Error messages, 2-20
<ESCAPE>, 5-8
Examine command, 4-29
Execute procedure (@), 4-32
Exit command, 4-34

F

FCS (file control services),
 1-4
File protection, 2-4
 specification of, 2-17
File specifications under
 VAX/VMS, 2-5
 defaults for, 2-8, 2-9
Files-11, 1-3
Files-11 Structure Levels 1
 and 2, 2-7
File types,
 defaults, 2-9
FLX (file transfer utility),
 1-3
 use under VAX/VMS, 3-9

G

Generic device names, 3-3

I

.IFACT directive, 5-6
.IFINS directive, 5-6
.IFLOA directive, 5-6
Image,
 definition of, 2-1
Image files,
 noncontiguous, 3-12
Images,
 running of, 3-6
Indirect command files, 5-1
 execution of, 3-10
 nesting of, 5-11
 switches, 5-7
Initialize command, 4-36
Installing RSX-11M images,
 3-5

L

Label table entries, 5-9
LB, 2-12
LBR (RSX-11M Librarian),
 1-2
Lexical functions, 5-12
Limits and quotas, B-1
Link command, 4-43
Logging into VAX/VMS, 3-1
Logical names, 2-10, 3-4
 alternative to Install,
 3-5
 alternative to Reassign,
 3-4
 alternative to Redirect,
 3-5
 assignment for RSX-11M
 images, 3-4
 group, 2-10
 process, 2-10
 system, 2-11
LOGIN.CMD file, 3-2
Login procedure, 4-49
Logout command, 4-50

M

MACRO-11 (MAC), 1-2
<MAPPED>, 5-8
<MEMSIZ>, 5-8
Mount command, 4-51
Mounting volumes, 3-3

N

Noncontiguous image files,
 3-12

INDEX (Cont.)

Numeric symbols, 5-8

O

ODT.OBJ, 1-3
On Control Y, 4-60
.ONERR directive, 5-7
On severity level, 4-62
On statement, 4-59
OV, 2-12

P

Password, 3-1
PAT (file patch utility),
 1-3
.PAUSE directive, 5-11
PIP, 1-3
Print command, 4-64
Priority, 2-5
Privileges, 2-4, A-1
Process,
 definition of, 2-1
 detached, 2-2
 login, 2-2
 Versus task, 2-1
Process identification, 2-2
Process name, 2-2
Process protection, 2-3, 2-4
Protection, 2-3, 2-4
 file, 2-4
 password, 3-1
 process, 2-4
 setting of, 4-87
Purge command, 4-60

Q

Quotas and limits, B-1

R

Radix indicator, 5-9
<RAD50>, 5-8
RMS-11, 1-4
RMS-11 utilities, 1-3
RSXMAC.MLB, 1-3
RSX-11M components,
 use of under VAX/VMS, 3-7
RSX-11M files available
 under VAX/VMS,
 ODT.OBJ, 1-3
 RSXMAC.MLB, 1-3
 SYSLIB.OLB, 1-3

RSX-11M images,
 requirements for execution,
 1-1, 1-2
RSX-11M program development
 utilities available
 under VAX/VMS, 1-2, 1-3
RSX-11M/S system generation,
 5-14
Run command, 4-69
Running an image (native or
 RSX-11M), 3-2

S

Selecting a command inter-
 preter, 3-2
Set commands, 4-79
 Set Card Reader, 4-81
 Set Control Y, 4-82
 Set Default, 4-83
 Set Magtape, 4-84
 Set On, 4-85
 Set Process, 4-86
 Set Protection, 4-87
 Set Queue, 4-88
 Set RMS Default, 4-89
 Set Terminal, 4-90
 Set /UIC, 4-92
 Set Verify, 4-93
 Set Working Set, 4-95
Show commands, 4-95
 Show, 4-95
 Show Daytime, 4-96
 Show Default, 4-97
 Show Devices, 4-98
 Show Logical, 4-99
 Show Magtape, 4-100
 Show Network, 4-101
 Show Printer, 4-102
 Show Process, 4-103
 Show Protection, 4-104
 Show Queue, 4-105
 Show RMS Default, 4-106
 Show Status, 4-107
 Show Symbols, 4-108
 Show System, 4-109
 Show Terminal, 4-110
 Show Translation, 4-111
 Show Working Set, 4-112
SLP (source language input
 program), 1-3
SP, 2-12
Special symbols, 5-8
Stop command, 4-113
<STRLEN>, 5-8
Subdirectories, 2-7
Submit command, 4-115

INDEX (Cont.)

Subprocess, 2-2
SY, 2-12
<SYDISK>, 5-8
Symbols, 5-7
 numeric, 5-8
 special, 5-8
 summary of, 5-8
Syntax rules, 2-15
SYSLIB.OLB, 1-3
System generation,
Pre-generation indirect
 command file, 5-14
RSX-11M/S, 5-14
<SYSUIC>, 5-8
SYS\$COMMAND, 2-11, 5-9
SYS\$DISK, 2-11
SYS\$ERROR, 2-11
SYS\$INPUT, 2-11, 5-9
SYS\$LIBRARY, 2-11
SYS\$OUTPUT, 2-11
SYS\$SYSTEM, 2-11
<SYUNIT>, 5-8

T

Task, 2-1
Terminal characteristics,
 summary of, 4-91
Terminal functions, 2-13 -
 2-15
TI, 2-12
Time command, 4-118
Times (absolute and delta),
 specification of, 2-18,
 2-19
TKB (RSX-11M Task Builder),
 1-2
 noncontiguous image files,
 3-12
 options, 3-12
 switches, 3-11
Type command, 4-119

U

<UIC>, 5-8
UFD command, 4-120
Unit numbers, 2-7
Unsupported directives, 5-6
User authorization file,
 2-3, 3-1
User identification code
 (UIC),
 group number, 2-3
 member number, 2-3
User name, 3-1

V

VAX/VMS components,
 use of from MCR, 3-10
Version numbers, 2-8
VFY1 (Files-11 Structure
 Level 1 verification
 utility), 1-3
Volumes, mounting, 3-3

W

.WAIT directive, 5-7
WK, 2-12

X

.XQT directive, 5-7

Z

ZAP utility, 1-3

READER'S COMMENTS

NOTE: This form is for document comments only. DIGITAL will use comments submitted on this form at the company's discretion. If you require a written reply and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Did you find this manual understandable, usable, and well-organized? Please make suggestions for improvement.

Did you find errors in this manual? If so, specify the error and the page number.

Please indicate the type of reader that you most nearly represent.

- ☐ Assembly language programmer
- ☐ Higher-level language programmer
- ☐ Occasional programmer (experienced)
- ☐ User with little programming experience
- ☐ Student programmer
- ☐ Other (please specify)_____

Name_____ Date_____

Organization_____

Street_____

City_____ State_____ Zip Code_____

or
Country

— — — Do Not Tear - Fold Here and Tape — — —

digital



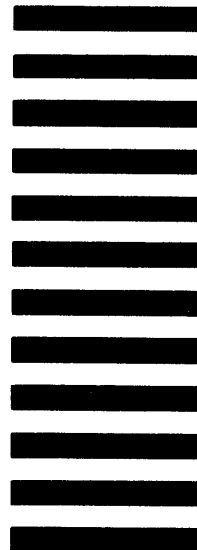
No Postage
Necessary
if Mailed in the
United States

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO.33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

RT/C SOFTWARE PUBLICATIONS TW/A14
DIGITAL EQUIPMENT CORPORATION
1925 ANDOVER STREET
TEWKSBURY, MASSACHUSETTS 01876



— — — Do Not Tear - Fold Here — — —