VAX-11/RSX-11M
User's Guide

Order No. AA-D037B-TE

March 1980

This document provides the information needed to use the VAX/VMS MCR command language, execute MCR indirect command files, and use RSX-11M Version 3.2 components under VAX/VMS.

VAX-11/RSX-11M
User's Guide

Order No. AA-D037B-TE

SUPERSESSION/UPDATE INFORMATION:	This revised document supersedes the VAX-11/RSX-11M User's Guide (Order No. AA-D037A-TE)
OPERATING SYSTEM AND VERSION:	VAX/VMS V02
SOFTWARE VERSION:	VAX/VMS V02

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

digital equipment corporation · maynard, massachusetts

First Printing, August 1978
Revised, March 1980

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

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

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

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

CONTENTS

FIGURES

TABLES

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 3.2 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 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 the VAX/VMS Summary Description and Glossary.

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 in to 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 to perform a RSX-11M/S system generation using VAX/VMS as a host system.

- Appendix A describes user privileges, and Appendix B lists resource limits and quotas.

ASSOCIATED DOCUMENTS

The following additional documents may also be useful:

- VAX-11 Information Directory and Index
- VAX-11/RSX-11M Programmer's Reference Manual
- VAX/VMS Primer
- VAX/VMS Command Language User's Guide
- VAX/VMS Guide to Using Command Procedures

Of the documents listed above, the VAX/VMS Command Language User's Guide provides the most useful additional information; since the VAX/VMS MCR command language and the primary VAX/VMS command language DCL have many common features. The VAX-11/RSX-11M User's Guide describes these features to the extent needed to perform the VAX/VMS equivalent of normal RSX-11M functions.

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.

SUMMARY OF TECHNICAL CHANGES

This manual has been revised to reflect the current support of RSX-11M Version 3.2 under VAX/VMS Version 2.0. Most of these changes refer to commands and their keywords, and indirect command files, as described, in Chapters 4 and 5, respectively. Changes made within this document are summarized below.

The following commands have been added to Chapter 4:

LIBRARY
MACRO
RUN (Image)
SET HOST
SET PASSWORD

New keywords with the commands that they modify are listed below as they appear in Chapter 4:

Command	**Keyword**
COPY	/VOLUME=n
DIRECTORY	/BEFORE[=time]
	/COLUMNS=n
	/CREATED
	/[NO]DATE[=option]
	/EXCLUDE=(file-spec[,...])
	/EXPIRED
	/[NO]HEADING
	/MODIFIED
	/OUTPUT[=file-spec]
	/[NO]OWNER
	/[NO]PROTECTION
	/SINCE[=time]
	/[NO]SIZE[=option]
	/TOTAL
	/[NO]TRAILING
	/VERSIONS=n
DMOUNT	/UNIT
INITIALIZE	/[NO]SHARE
LINK	/[NO]CONTIGUOUS
	/HEADER
	/P0IMAGE
	/PROTECT
	/[NO]SHAREABLE[=file-spec]
	/[NO]SYSTEM
	/[NO]USERLIBRARY[=(table [,...])]
PRINT	/CHARACTERISTICS=(c[,...])

PURGE	/[NO]LOG
SUBMIT	/CPUTIME=n /REMOTE /WSDEFAULT=n /WSQUOTA=n

The following directives, which have been added to the indirect command file processor, are documented in Table 5-1:

.BEGIN

.ENABLE QUIET

.END

.ERASE GLOBAL

.ERASE LOCAL

.EXIT [n]

.ONERR Label

.OPENA [#n] file-spec

.STOP [n]

The following special symbols have been added and are documented in Table 5-2:

<DATE>

<EXSTAT>

<LIBUIC>

<RSX11D>

<STRLEN>

<SYSTEM>

<TIME>

Figure 5-1 in Chapter 5 documents changes to the Pre-System Generation Indirect Command File.

The following types of miscellaneous textual changes have been made:

Version numbers have been changed to denote the current versions of VAX/VMS and RSX-11M.

Cross-references have been added to facilitate use of this manual.

CHAPTER 1

INTRODUCTION

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

- DIGITAL command language (DCL), as introduced in the VAX/VMS Primer, defined in the VAX/VMS Command Language User's Guide, and whose command procedures are discussed in the VAX/VMS Guide to Using Command Procedures.

- MCR command language, as described in this document and referred to in the VAX-11/RSX-11M Programmer's Reference Manual.

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 3.2 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 EXECUTABLE IMAGE TYPES

VAX/VMS MCR provides the tools needed to develop and run both RSX-11M task images and native VAX-11 images. This 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 be run from one indirect command file using MCR.

However, for support of MCR indirect command files under VAX/VMS, your command interpreter must be MCR. For information on how to establish MCR as your command interpreter, refer to Section 3.1.1, "Selecting a Command Interpreter."

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 processor and the VAX/VMS operating system impose restrictions on the RSX-11M images that can execute. The restrictions at both hardware and software levels, which are described in the VAX-11/RSX-11M Programmer's Reference Manual, are summarized below.

1.1.1 Requirements for RSX-11M Image Execution

The processor requirements for RSX-11M image execution are as follows:

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

The software requirements are as follows:

- The image must have been built using the RSX-11M Task Builder to execute in a mapped system.
- The image must not depend on environmental features of RSX-11M that are not available in VAX/VMS, for example, PLAS memory management or significant events.
- The image must not physically overmap the RSX-11M executive; the RSX-11M executive is not present in VAX/VMS.
- The image must not physically overmap the PDP-11 I/O page; the I/O page is not available in user mode.
- The image must not use any DECnet-11 functions.

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 Task Builder before they can execute under VAX/VMS.

RSX-11M 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 which allow processing of VAX/VMS file specifications.

1.2 RSX-11M PROGRAM DEVELOPMENT TOOLS

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)
- Line Text Editor (EDI)
- RSX-11M Task Builder (TKB)
- RSX-11M Librarian (LBR)
- 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.SML [1,1]
- ODT.OBJ

With the utilities and files listed above, you can use VAX/VMS as you would use 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 Version 3.2.

1.2.2 Additional Facilities

The following RSX-11M facilities can be performed under VAX/VMS:

- RSX-11M/S system generation
- Operation of PDP-11 language compilers

Note that in order to obtain the necessary components, a separate license is required for each.

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 Files-11 Structure Level 1, whereas VAX/VMS supports both structure levels. By default, VAX/VMS creates Files-11 Structure Level 2 volumes; however, it can create and read Files-11 Structure Level 1 volumes. Thus, Files-11 Structure Level 1 volumes can be transported among RSX-11, IAS, and VAX/VMS systems.

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 QIO requests from the RSX-11M format to VAX/VMS equivalent native mode QIOs. The VAX-11/RSX-11M Programmer's Reference Manual describes the conversion process.

However, RMS-11 record locking is not supported by RSX-11M MCR.

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. Most directives are supported by VAX/VMS; however, the following are not:

- PLAS directives, because of substantial differences in hardware memory management techniques
- GET SENSE SWITCH, because VAX-11 does not have sense switches
- CONNECT TO INTERRUPT VECTOR, because I/O drivers do not run in compatibility mode under VAX/VMS.
- CONNECT, because VAX/VMS inherently has more process protection than RSX-11M.

1.6 SUMMARY OF COMMANDS

Tables 1-1 through 1-5 summarize the VAX/VMS MCR commands. In addition to the commands listed, VAX/VMS MCR recognizes commands that invoke RSX-11M components. For example, MAC invokes the MACRO-11 assembler.

Table 1-1
Summary of MCR Initialization Commands

Command	Function
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
Dmount	Releases a volume previously specified in a Mount command
Initialize	Initializes a Files-11 Structure Level 1 or 2 disk volume
Mount	Makes a disk or magnetic tape volume or volume set available for processing
Set Host	Establishes a virtual communication link between a terminal and a network node to which the terminal is not directly connected
Set Password	Allows users to change their own password
Set UIC	Changes the default directory and the user identification code (UIC) of the current process

Table 1-2
Summary of MCR Informational Commands

Command	Function
Time	Displays the current date and time

Table 1-3
Summary of MCR Process Control Commands

Command	Function
Bye	Terminates an interactive terminal session
Cancel	Cancels scheduled wake-up requests for a process
Continue	Resumes execution of an interrupted image or an indirect command file

(continued on next page)

Table 1-3 (Cont.)
Summary of MCR Process Control Commands

Command	Function
Exit	Terminates processing of the current indirect command file
On	Defines the default course of action to be taken when a command or program in an indirect command file encounters an error condition of a particular severity-level or when CTRL/Y is used
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

Table 1-4
Summary of MCR Program Development Commands

Command	Function
Debug	Invokes a debugger for an interrupted image
Deposit	Replaces the contents of specified locations in virtual memory
Examine	Examines the contents of specified locations in virtual memory
Library	Invokes the native mode librarian
Link	Links native VAX-11 images
Macro	Invokes the VAX-11 MACRO assembler

Table 1-5
Summary of MCR File Manipulation Commands

Command	Function
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
Delete	Deletes files, entries from print 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
UFD	Creates a directory or subdirectory on a Files-11 Structure Level 1 or 2 disk

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 in to 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; Peripheral Interchange Program (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 and Glossary 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 whom 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, which is the process created for you when you log in to the system

- A detached process, which is a fully independent process

- A subprocess, which is a process that you create to run a specified image, but which remains under control of your login 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, in which 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 in which 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 whose first 3 characters do not form an ellipsis (...) in its image label block. Utilities do not cause the process name to change because these names start with an ellipsis, (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 that does not start with an ellipsis; the task name becomes the process name for the duration of that image. When the image terminates, VAX/VMS restores the previous name of the process.

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 exists.

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

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
- User identification code (UIC)
- User account name
- User default disk device
- User default command language interpreter (CLI)
- 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 base priority
- Amount of resources available to user

When you log in, VAX/VMS uses your user 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 numeric 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.

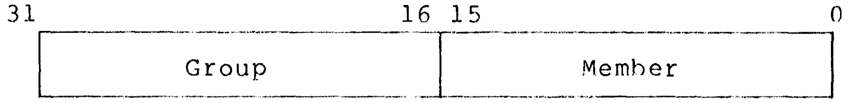

Figure 2-1 Format of VAX/VMS UICs

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 that 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 that 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 privileges differ from RSX-11M privileges in two respects:

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

- Users are not categorized as privileged or nonprivileged; rather, the VAX/VMS system manager can associate any of a wide range of privileges with each user, depending on that 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.

During execution, if an image attempts to use a function for which the user does not have the privilege, the image 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 (Group) and world (World) process control. 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 in the system 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 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 own 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. 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]file-name.type;ver
```

node — Node name. The node name is a 1- to 6-character alphanumeric string which must contain at least one alphabetic character. The node name identifies a location on the network and is separated from the device name by two colons (::).

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 maximal length of the device name field is 15 characters.

The device name is separated from the directory name by a colon (:).

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 Files-11 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 square 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 card characters in the file name, file type, and version number fields of a file specification. For further information concerning wild card characters refer to Chapter 2 in the VAX/VMS Command Language User's Guide.

2.3.1 Device Names

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

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

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

Usually drivers used for any of the devices listed above are loaded during system generation. However, if a driver is needed but it has not been loaded, ask the system manager for assistance.

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.

Table 2-1
VAX/VMS Device Types

Mnemonic	Device Type
CR	Card reader
CS	Console Storage Device
DB	RP04, RP05, RP06 disk
DD	TU58, Cassette Tape
DL	RL02, Cartridge Disk
DM	RK06, RK07 Cartridge Disk
DR	RM03 Disk
DY	RX02 Floppy Diskette
LA	LPA11-K Laboratory Peripheral Accelerator
LP	Line Printer
MB	Mailbox
MS	TS-11 Magnetic Tape
MT	TE16, TU45, TU77 Magnetic Tape
NET	Network Communications Logical Device
OP	Operator's Console
RT	Remote Terminal
TT	Interactive Terminal
XF	DR32 Interface Adapter
XJ	DUP11 Synchronous Communications Line
XM	DMC11 Synchronous Communications Line

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 as many as 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.

Use the MCR UFD command as described in Chapter 4 to create directories on a Files-11 Structure Level 2 or Files-11 Structure Level 1 volumes.

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 contents of a directory or subdirectory.

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	The local system is your default.
device	The disk is 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	The directory is established as your default by (1) your entry in the user authorization file or (2) execution of a Set Default or the Set UIC command.
file name	There are 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.

Table 2-3
Default File Types

File Type	File Contents
ANL	Output file for the Analyze command
BAS	Source input for the BASIC-PLUS and VAX-11 BASIC compiler
B2S	Source input for the PDP-11 BASIC-PLUS 2/VAX compiler
B32 or BLI	Source input for the VAX-11 BLISS-32 compiler
CBL	Source input to the PDP-11 COBOL-74/VAX compiler

(continued on next page)

Table 2-3 (Cont.)
Default File Types

File Type	File Contents
CMD	RSX-11M indirect command file to be executed with the @ (Execute Command Procedure) under the MCR command interperter
COB	Source input for the VAX-11 COBOL-74 compiler
COM	VAX/VMS command procedure file that is executed with the @ (Execute Procedure) command, or submitted for batch execution with the Submit command under the DCL command interperter.
COR	Source input file for the VAX-11 CORAL 66 compiler
DAT	Input or output data file
DIF	Output listing created by the Differences command
DIR	Directory file
DIS	Distribution list for the Mail command
DMP	Output from the Dump command
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 compiler
FTN	Source language input for the PDP-11 FORTRAN IV or FORTRAN IV-PLUS compiler
HLB	Help text library file
HLP	Input source file for help libraries
JNL	Journal file output from PATCH utility
JOU	Journal file/audit trail from EDT
L32	Precompiled Library for VAX-11 Bliss-32
LIB	Input file containing VAX-11 COBOL-74 source Statements
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

(continued on next page)

Table 2-3 (Cont.)
Default File Types

File Type	File Contents
MAI	Mail message file
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
MSG	Source file that specifies the text of messages
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
PAR	A SYSGEN parameter file
PAS	Source input for the VAX-11 PASCAL compiler
R32 or REQ	VAX-11 BLISS-32 source file required for compilation
STB	Symbol table file created by the VAX-11 Linker or the RSX-11M Task Builder
SYS	System Image
TLB	Text library
TMP	Temporary file
TXT	Input file for text libraries or output file for the Mail command
UPD	Update file of changes for a VAX-11 MACRO source program

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 the following:

- A physical-device, logical-device, or pseudodevice name

- A file specification or portion of one

Section 3.2.3 describes the use 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 a 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 equivalence names 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. Thus, 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 logging 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 they are explicitly deleted.

The GRPNAM privilege is required for a user to place a name in or delete a name from 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 manner in which entries in this table are used is similar to the manner in which global assignments are used under RSX-11M. Entries remain in the table until they are explicitly deleted.

The SYSNAM privilege is required for a user to place a name in or delete a name from 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 assigned to the terminal. For a batch job, SYS$INPUT is assigned to the batch input stream.

- SYS$OUTPUT -- command output stream for the process. For an interactive user, SYS$OUTPUT is assigned to the terminal. For a batch job, SYS$OUTPUT is assigned to the batch job log file. For indirect command files, you can temporarily assign SYS$OUTPUT to a different file as described by the Execute Procedure in Chapter 4.

- SYS$COMMAND -- original SYS$INPUT for a process. 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 assigned to the terminal. For a batch job, SYS$ERROR is assigned to the batch job log file.

- SYS$DISK -- default disk device established at login or established by the Set Default command described in the VAX/VMS Command Language User's Guide.

- SYS$LOGIN -- created at login time to refer to the device and directory of the login process which is specified in each user authorization file.

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$HELP -- the device and directory of system help files.

- SYS$LIBRARY -- the device and directory that contains system libraries.

- SYS$MESSAGE -- the device and directory of system message files.

- SYS$NODE -- the device and directory names of the current network node for the local system, if DECnet is active on the system.

- SYS$SHARE -- the device and directory of system shareable images.

- SYS$SYSDISK -- the device and directory of the VMS system disk; that is, the device and directory on which SYS$SYSTEM resides.

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

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

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 pseudodevice 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 that uses an RSX-11M device name.

2.5.1 Mapping RSX-11M Pseudodevice Names

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

Table 2-4
Mapping of RSX-11M Pseudodevice Names

RSX-11M Pseudodevice 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. The result is the controller letter. The remainder becomes the VAX/VMS unit number. For example, RSX-11M devices TT0 and DB18 become VAX/VMS devices TTA0 and DBB2, respectively, as follows.

TT0 to TTA0:

```
     Controller and unit ='A'+(0/16)='A'+0 with a remainder of 0
                           'A'+0 ='A'= controller
                               0 = unit number
```

DB18 to DBB2:

```
     Controller and unit = 'A'+(18/16)='A'+1 with a remainder of 2
                           'A'+1='B'= controller
                               2 = unit number
```

VAX/VMS performs this conversion when assigning an I/O device to 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.

Table 2-5
Terminal Function Keys

Key	Function
RETURN	(Carriage return) Transmits the current line to the system for processing. Before a terminal session, initiates login sequence.
Control characters:	Define functions to be performed. You enter a control character by depressing the CTRL key while you press a letter key. All CTRL/x key sequences are echoed on the terminal as ^x.
CTRL/C	Before a terminal session, initiates a 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 the 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)

Table 2-5 (Cont.)
Terminal Function Keys

Key	Function
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 position 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 for 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:

- 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 wherever 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. For example:

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

No restriction is placed on the number of lines you can use to enter a command string.

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

2.7.2 Using Comments in 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

Most 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

However, you must type the first four characters of some commands in order for VAX/VMS MCR commands to be distinguished from RSX-11M command names which contain the same initial three characters. These exceptions are noted in the command descriptions in Chapter 4.

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 Placing Keywords

VAX/VMS MCR is 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 whether they appear after the command name or after 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 after a file specification, they affect only the file thus qualified. If specified after 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, 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 the 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 0 through 377 (octal) and users with physical I/O or logical I/O privilege (generally, system managers, system programmers, and operators). The default for group numbers for system users is 0 through 10 (octal).
- 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.

When you specify a protection code, separate each user category from the access type with a colon. If you want 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 each category with a comma and enclose the entire code specification 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 generally have the following format.

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

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

Field	**Meaning**
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	Seconds (0 through 59)
c	Hundredths of seconds (0 through 99)

You may also specify one of the following keywords whenever an absolute time is appropriate:

Keyword	**Description**
TODAY	The current day, month and year at 00:00:00:0 o'clock
TOMMORROW	24 hours after 00:00:00:0 o'clock today.
YESTERDAY	24 hours before 00:00:00:0 o'clock today.

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

- If you specify both the date (dd-mmm-yyyy) and the time (hh:mm:ss.c), you must type a 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 format must contain at least one hyphen.
- 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.
- The period between seconds and hundredths of seconds is a delimiter; it is not a decimal point.

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:

Time Specification	**Result**
28-JUN-1980:12	12:00 noon on June 28, 1980
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.c]
```

The variable fields are as follows:

Field	**Meaning**
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	Number of seconds
c	Number of hundredths of seconds (0 through 99)

When you specify a delta time value, you can truncate the time field on the right; you can 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:

Time Specification	**Result**
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 that they issue under RSX-11M. When using VAX/VMS MCR to run RSX-11M components, you should consult the appropriate RSX-11M 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 or the Linker)
- Execute RSX-11M indirect command files (for example, RSX-11M/S system generation indirect command files)

This chapter explains how to log in to 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 IN TO VAX/VMS

To log in to the VAX/VMS system, you must first gain the attention of login. To do this, type 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 a login indirect command file is not specified there, login searches your default directory for an indirect command file named LOGIN.CMD and executes this file if it is present. Finally, the 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 that you specify after your user name to override the default command interpreter for the terminal session

If the MCR command interpreter is not your customary default, then you must enter your user name followed by /CLI=MCR, as follows.

```
Username: CALEB/CLI=MCR
```

Then, enter your password in the normal manner.

Note that the VAX/VMS system does not support MCR indirect command files unless your command interpreter is MCR.

Also, once you log in with MCR as your command interpreter it remains in control of your process until you log out. Therefore, DCL commands and the DCL command interpreter cannot be accessed when you are logged in with the VAX/VMS MCR command interpreter.

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. The LOGIN.CMD must contain valid VAX/VMS MCR commands and directives.

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 image that runs in your current process or in 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 explicitly specifies the device type and, optionally specifies the controller or unit. A generic-device name allows the system to select the actual device unit to be allocated from available devices of a specific type. For example, the following command requests VAX/VMS to allocate any available TE16 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 a specific 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 specific device unit available.

Unlike RSX-11M, when the controller and unit number are omitted, 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; refer to Chapter 5 in this manual.

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)

All of the Mount command options are described in Chapter 4 in the Mount command description.

The main purpose of the Mount command is to gain access to the volume. However, under VAX/VMS you can use certain disk volumes without issuing a Mount command for them. Any volume that another user (most likely, the system manager or operator) has mounted specifying the /SYSTEM keyword is available to anyone logged in to 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, you can do so either explicitly or by using 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:
```

You also 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 logical names, VAX/VMS logical names are not restricted to two letters and a unit number. VAX/VMS logical names can consist of as many as 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 has 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 before 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 capability. 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; Chapter 2 in the VAX/VMS Command Language User's Guide provides detailed information on logical names.

3.2.4 Installing RSX-11M Images

Under VAX/VMS, images do not have to be 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 that of the Install command. If you assign a logical name to the file specification of the image file, the logical name serves as a task name. If you type the logical name in response to the MCR prompt (>), VAX/VMS performs logical name translation and runs 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.

Note that for a process to place names in the system logical name table the user must have the SYSNAM privilege.

3.3 RUNNING IMAGES

The VAX/VMS MCR 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 quotas, and privileges. A detailed description of the Run command and all its options is in Chapter 4.

The following are examples of the Run command:

```
     > RUN INVENTORY
```

This command 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
```

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

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.

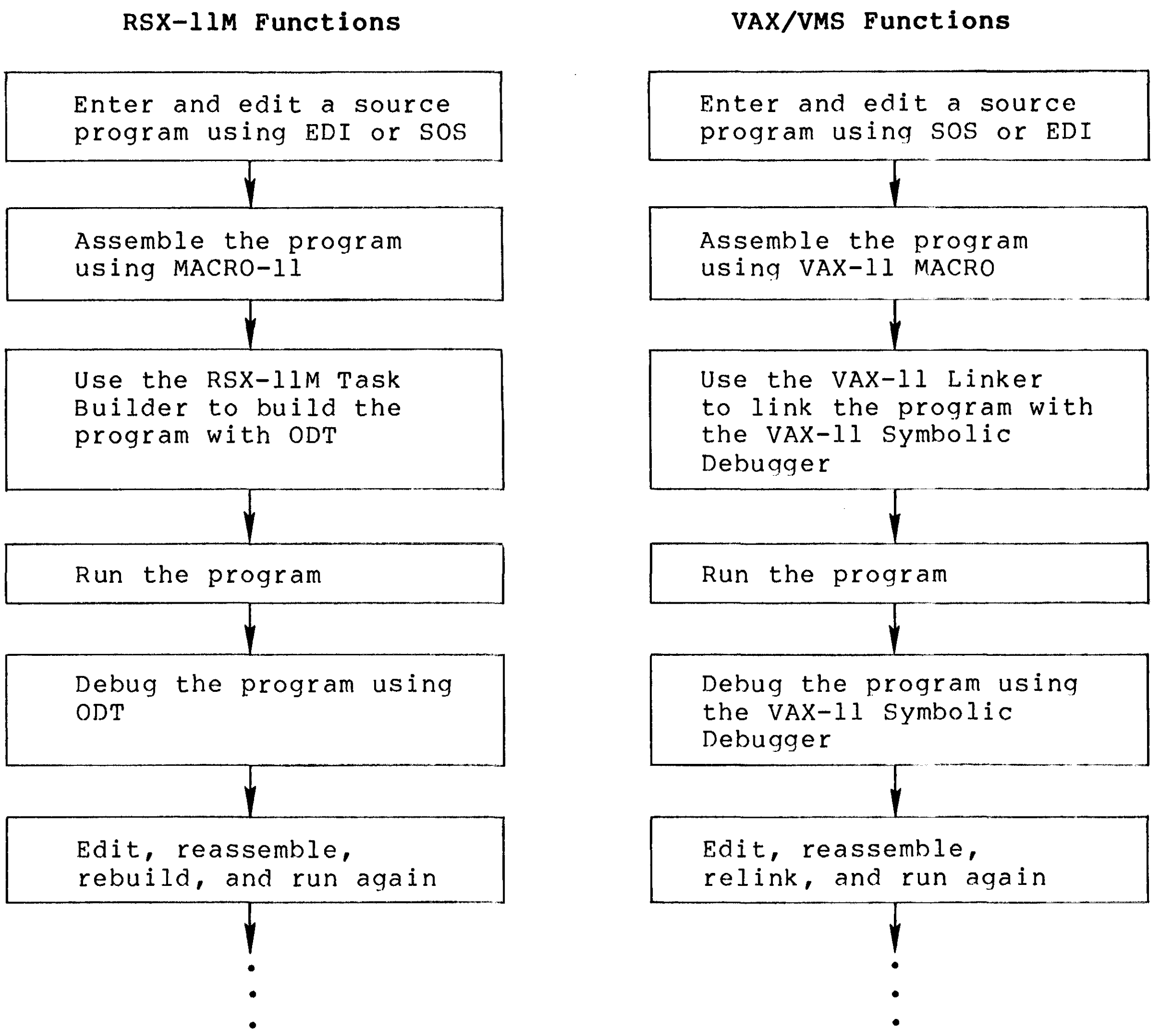
RSX-11M Functions
VAX/VMS Functions
Enter and edit a source program using EDI or SOS
Enter and edit a source program using SOS or EDI
Assemble the program using MACRO-11
Assemble the program using VAX-11 MACRO
Use the RSX-11M Task Builder to build the program with ODT
Use the VAX-11 Linker to link the program with the VAX-11 Symbolic Debugger
Run the program
Run the program
Debug the program using ODT
Debug the program using the VAX-11 Symbolic Debugger
Edit, reassemble, rebuild, and run again
Edit, reassemble, relink, and run again

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 if the image was built with ODT.

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-11 Utilities Reference Manual describes operation of VFY1.

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

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

```
> EDI
EDI> DBA1:[COLD]CENTIGRAD.DAT
[CREATING NEW FILE]
INPUT
  .
  .
  .
<RET>
*EX
[EXIT]
>
```

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 CENTIGRAD.DAT. Failure to locate the file results in creation of a new file under the directory [COLD]. Exiting from EDI causes MCR to prompt.

```
> MAC @MACINP
>
```

This command 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.

```
> 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 which passes control to the MCR command interpreter.

3.4.2 Using the File Transfer Program

The File Transfer Program (FLX) interprets only RSX-11M file specifications, that is, file specifications with the following format:

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

FLX cannot interpret a controller designation in the device name or a directory name (other than 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.

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/DO
```

VAX/VMS maps the device name DB0 to physical device DBA0. It maps device name MT17 to device MTB1. File SYS1.MAC is transferred to directory [120,30] on DB0.

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

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.

See the VAX-11 Utilities Reference Manual for further information on FLX.

3.4.3 Using VAX/VMS Components

From the VAX/VMS MCR command interface, you can also invoke the following native VAX/VMS components listed below. The information in parentheses indicates the component or command file used in requesting each component.

- VAX-11 MACRO (Macro command)
- Standard VAX-11 Editor (SOS)
- VAX-11 Librarian (Library command)
- VAX-11 Linker (Link command)
- Various VAX-11 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 3-character component name or the 3-character component name followed by a valid command line for that component
- The command that invokes the particular component

Subsequent operation of the requested component is described in the appropriate VAX/VMS document; see the VAX-11 Information Directory and Index 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 command files specified for use by RSX-11M components (for example, Task Builder and MACRO-11) are identical to those used under RSX-11M. Section 3.1.1, "Selecting a Command Interpreter," describes the login procedure used to establish MCR as your command interpreter; which is necessary for support of indirect command files. Chapter 5 describes the MCR indirect command file processor directives available under VAX/VMS and provides additional information about running MCR indirect command files. The Execute Procedure (@) is described in Chapter 4.

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.

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 user identification code (UIC), priority, and partition have no effect; UIC and priority are associated with the user running the image under 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. 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-device name or a 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.

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	Postmortem dump is requested
SH	Short memory allocation file is requested
SP	Spool map output
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

CHAPTER 4

MCR COMMANDS

This chapter describes VAX/VMS MCR commands and their keywords in detail. Individual command descriptions note restrictions between VAX/VMS MCR commands and RSX-11M MCR commands. For general information on the VAX/VMS MCR environment, refer to Chapter 2 which provides a summary of the VAX/VMS MCR environment and highlights the differences between VAX/VMS MCR and RSX-11M MCR.

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:

```
     > ALLOCATE 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 described in Section 4.3 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.

- The form of the Allocate command that is compatible between VAX/VMS and RSX-11M systems is:

```
          ALLOCATE device-name[:]
```

4.2 APPEND

The Append command adds the contents of one or more specific input files to the end of a specific 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 character 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 each file specification with either a comma (,) or a plus sign (+).

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

Command Keyword:

/LOG
/NOLOG
(default)

Controls whether the file specification of each file that is appended is displayed.

If you specify the /LOG keyword with the Append command, the file specifications of the input and and output files and either the number of blocks or records that are appended will be displayed. The number of new files created will be displayed at the end of command processing.

File Keywords:

/ALLOCATION=n

Forces the initial allocation of the new output file to the number of blocks specified by n. The value n is interpreted as a decimal number by default.

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

/CONTIGUOUS
/NOCONTIGUOUS
(default)

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 keyword is valid only for new relative files. If you specify the /FILE-MAXIMUM keyword, the /NEW keyword is assumed.

/NEW
/NONEW (default)

Requests that, if a specific output file does not already exist, the Append command create one. Because /NONEW is the default, the output file must already exist.

/PROTECTION=code

Defines the protection to be applied to the output file. The protection is specified following the standard rules, described in Section 2.7.5. Any protection attributes not specified are taken from the current protection of the output file or, if a new file is created, from the current default protection.

/READ_CHECK
/NOREAD_CHECK
(default)

Indicates whether the Append command is to read a specific input file(s) twice to verify that all records have been correctly read.

/WRITE_CHECK
/NOWRITE_CHECK
(default)

Indicates whether the Append command is to read the output file after it has been written. This verifies 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 use wild card characters to concatenate Files-11 Structure Level 1 files, the input files are copied to the output file in random order. When you use wild card characters to concatenate Files-11 Structure Level 2 files, the Append command copies files in alphanumeric order. When wild card characters 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 and information about wild card characters 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-device, logical-device, or pseudodevice) name to which the logical name is to be assigned. The equivalence name can be from 1 through 15 characters.

nn Specifies the device unit number.

ll Specifies a logical name from 1 through 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 (default) 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.

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, described in the VAX/VMS Command Language User's Guide, can be used to display logical names in the process, group, and system 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 there are any.

Format:

```
BYE
```

Example:

```
> BYE
  MURPHY    logged out at 4-APR-1980 13:33:26.57
```

Note:

- The Logout command performs the same function as the Bye command.
- RSX-11M supports the Bye command.

4.5 CANCEL

The Cancel command requests the system to cancel a wake-up request for a specific process.

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

- The RUN$ directive (see Notes below)
- The Schedule Wake-up 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=process-id

process-name — Specifies the name of the process for which wake-up 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=process-id

Specifies the process identification of the process for which wake-up 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-up requests for your current process are canceled. The process-id value is interpreted as a hexidecimal number.

Examples:

```
> CAN PROCA
```

This command cancels wake-up requests for a process executing in the same group as the requesting process. If the process is not a subprocess of the requesting process, then group privilege is required.

```
> CAN /IDENTIFICATION=C0025
```

This command cancels wake-up requests for the process whose process identification number is C0025. It can be executing outside the group requesting process; appropriate privilege is required, however.

Notes:

- You must have group or world privilege to cancel wake-up 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 wake-up 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 described in the VAX/VMS Command Language User's Guide.

- VAX/VMS uses the Schedule Wake-up system service to emulate the RSX-11M RUN$ directive; for additional information, refer to the VAX-11/RSX-11M Programmer's Reference Manual.

- RSX-11M supports the Cancel command when the VAX/VMS MCR process name is equal to an RSX-11M task name.

4.6 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 if the program is to continue to execute. 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 Chapter 5.

Format:

```
CONTINUE
```

Example:

```
> RUN TEST
     .
     .
     .
  ^Y
> TIME
  4-APR-1980  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. This 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
                   = 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
/VOLUME=n
/[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 character in any field of the output file specification, the Copy command uses the related field of the input file specification(s) and does not concatenate the output. For more information on wild card characters, refer to the chapter on file specifications in the VAX/VMS Command Language User's Guide.

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 wild card characters in place of the file name, type, or version field. Then, all files that satisfy the remaining components are copied. For more information on wild card characters, refer to the chapter on file specifications in the *VAX/VMS Command Language User's Guide*.

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 card characters 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 (default)
/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 file is contiguous, use the /ALLOCATION and /CONTIGUOUS keywords.

The /CONTIGUOUS keyword has no effect when you copy files to or from tapes because the size of the input file cannot be determined until the file is read. 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 keyword 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 non-file-structured device.

/PROTECTION=code

Defines the protection to be applied to the output file. The protection is specified following the standard rules, described in Section 2.7.5. 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 a specific 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 specific file(s) are copied to the output file(s).

/VOLUME=n

Requests that the Copy command place the entire output file on the specified relative volume of a multivolume set.

If the /VOLUME keyword is not specified, the file is placed in an arbitrary position within the multivolume set.

/WRITE_CHECK
/NOWRITE_CHECK
(default)

Indicates whether the Copy command is to read the output file after it has been written to. This verifies that the file has been successfully copied and that the output file can be read without error.

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 RSX-11M, you can use PIP instead of Copy.

- When you use wild card characters 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 card characters 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 and further information on wild card characters.

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 name or logical name assigned to the device to be deallocated. If you omit the controller designator and unit number of 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:

```
> DEALLOCATE MT0
```

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

Note:

- The compatible form of the Deallocate command supported by both RSX-11M and VAX/VMS systems is:

  ```
  DEALLOCATE device-name:
  ```

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
```

Examples:

```
> 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, deletes entries from a print queue or a batch queue, and stops the printing of a file.

You specify /ENTRY to delete a batch or print 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
/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 each 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 card characters. Subsequent file specifications can omit any of these fields except the version numbers; the first file specification provides defaults for additional file specifications.

Command Keywords:

/BEFORE[=(abs-time)]
/BEFORE[=TODAY]
/BEFORE[=YESTERDAY]

Chooses only those specific files that have a selection date before the specified date. The selection date can be specified as an absolute date and time or as TODAY or YESTERDAY. Section 2.7.6 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.

/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 a T or a Y followed by RETURN in response to the message, the file is deleted. Any other response causes the file to be retained.

/CREATED

Requests that specific file(s) 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.

/EXPIRED

Selects specific 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 specific 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[=TODAY]
/SINCE[=YESTERDAY]

Chooses only those specific files that have a selection date after the specified date. The selection date can be specified as an absolute time or as TODAY or YESTERDAY. Section 2.7.6 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 TODAY by default; that is, the command deletes all specified files with a selection date of today.

Example:

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

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

Note:

- RSX-11M does not support this command.

DELETE/ENTRY

4.11 DELETE/ENTRY

The Delete/Entry command deletes one or more entries from a print or batch job queue. The /ENTRY keyword is required.

The job(s) to be deleted must have been queued by the current process or any process in the same group as the current process.

You can delete jobs that have not yet begun processing or files that are currently being processed.

When you need to delete an entry from a batch queue while the job is running, use the Stop/Entry command. To delete a print job that is running, consider either the Stop/Abort or the Stop/Requeue commands.

The /ENTRY keyword requires at least one job-number parameter to specify the job number(s) of one or more jobs to be deleted from a print or batch job queue. If you specify more than one job number, separate the numbers with commas and enclose the list in parentheses.

Format:

```
DELETE/ENTRY=(job-number[,...]) queue-name[:]
```

queue-name[:]   Specifies the name of the queue in which the job(s) exist.

Examples:

```
>PRINT/HOLD   ALPHA.TXT
 Job 110 entered on queue SYS$PRINT
  .
  .
  .

>DELETE/ENTRY=110  SYS$PRINT
```

The Print command queues a copy of the file ALPHA.TXT in a hold status to defer its printing until a later time. The system displays the job number 110 and the name of the queue in which the file was entered. Later, the Delete/Entry command requests that the entry be deleted from the queue SYS$PRINT.

```
>SUBMIT/HOLD/PARAMETERS=SCANLINE  DOFOR
Job 203 entered on queue SYS$BATCH
>SUBMIT/AFTER=18:00  WEATHER
Job 210 entered on queue SYS$BATCH
  .
  .
  .

>DELETE/ENTRY=(203,210) SYS$BATCH
```

The Submit commands spool the command procedures DOFOR.COM and WEATHER.COM for processing as batch jobs. DOFOR.COM is queued in a hold status; WEATHER.COM is queued for execution after 6:00 P.M. Later, The Delete/Entry command requests that both these entries be deleted from the queue SYS$BATCH.

4.12 DEPOSIT

The Deposit command replaces the contents of a specific 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] location=data [,...]
```

keywords

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

location

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.

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 location 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.

```
> .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 Chapter 5 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

4.13 DIRECTORY

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

The output of the Directory command depends on certain formatting keywords and their defaults. These keywords are: /COLUMNS, /DATE, /FULL, /OWNER, /PROTECTION, and /SIZE. However, Files-11 Structure Level 2 files usually are listed in alphabetical order, with the highest-numbered versions first. Files-11 Structure Level 1 files are listed randomly. The page width is adjusted automatically to the number of columns requested.

In studying the keywords and the capabilities they offer, watch for keywords that work together as well as keywords that override other keywords. For example, if you want the full format, you cannot expect that much information in one column. Therefore, if you specify both /COLUMNS and /FULL, the number of columns you request is ignored.

Format:

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

keywords

```
/BEFORE[-time]
/BRIEF
/COLUMNS=n
/CREATED
/[NO]DATE[=option]
/EXCLUDE=(file-spec[,...])
/EXPIRED
/FULL
/[NO]HEADING
/MODIFIED
/OUTPUT[=file-spec]
/[NO]OWNER
/PRINTER
/[NO]PROTECTION
/SINCE[=time]
/[NO]SIZE[=option]
/TOTAL
/[NO]TRAILING
/VERSIONS=n
```

file-spec[,...]

Specifies one or more files to be listed. The syntax of a file specification determines what file(s) will be listed, as follows:

- If you do not enter a file specification, the Directory command lists all versions of the files in your current default directory.

- If you specify only a device name, the Directory command uses your default directory specification.

- 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 all versions.
- If a file specification contains only a file name, the Directory command assumes all file types and versions.

If you specify more than one file, separate the file specifications with either commas (,) or plus signs (+). You can use wild card characters in the directory specification, file name, file type, or version number fields of a file specification to list all files that satisfy the components you specify. See the chapter on file specifications in the VAX/VMS Command Language User's Guide for a full description of wild card characters.

Command Keywords:

/BEFORE[=time]

Specifies that only those files dated earlier than a particular time be printed. You can specify one of the following values:

absolute-time — An absolute date and time. Observe the syntax rules for date and time values specified in Section 2.7.6.

TODAY — The absolute date and time representing the current day, month, and year, at 00:00:00.0 o'clock.

TOMORROW — The absolute date and time representing 24 hours after 00:00:00.0 o'clock today.

YESTERDAY — The absolute date and time representing 24 hours before 00:00:00.0 o'clock today.

This keyword is normally used in conjunction with one of the following keywords: /CREATED, /EXPIRED, or /MODIFIED. If you omit the /BEFORE keyword, you obtain all the files created, regardless of date. However, if you specify /BEFORE without a date or time, the default provides the files created prior to today.

/BRIEF (default)

Includes only the file name, type, and version number of each file to be listed. The default ouput format is /BRIEF. However, the /BRIEF keyword is overridden, whether specified explicitly or by default, whenever any of the following formatting keywords are specified in the command: /SIZE, /DATE, /OWNER, /PROTECTION, or /FULL.

The brief format lists the Files-11 Structure Level 2 files in alphabetical order from left to right on each line, in descending version number order. Files-11 Structure Level 1 files are listed randomly.

/COLUMNS=n

Lists the files using the specified number of columns on each line of the display. This keyword is used in conjunction with the /BRIEF keyword (either explicitly or by default). By default, the number of columns in the brief format is four; however, you may request the brief format with as many columns as you desire. When other formatting keywords are specified in the command, they override the /COLUMNS keyword.

/CREATED (default)

Selects the files according to their date of creation. This keyword is relevant only when used with the /BEFORE or /SINCE keywords, and should not be used with the /EXPIRED or /MODIFIED keywords. By default, when files are selected according to a particular date and time, the creation date is always used.

/DATE[=option]
/NODATE (default)

Includes the creation, expiration, or date last written for each file listed. If you omit this keyword, the default is /NODATE. However, if you specify /DATE without an option, the creation date is provided.

You may specify one of the following options with the /DATE keyword:

ALL

Lists all three file dates in the order, left to right, CREATED, MODIFIED, EXPIRED.

CREATED

Lists the creation date with each file.

EXPIRED

Lists the expiration date with each file.

MODIFIED

Lists the last date the file was written.

/EXCLUDE=(file-spec[,...])

Excludes the listed file specification(s) from the directory search. You may use wild card characters, in file specification(s) as described in the chapter on file specifications of the VAX/VMS Command Language User's Guide. At least one file specification is required for this keyword, but the file specification must not include a device or directory specification. Separate multiple file specifications by commas, and enclose the list in parentheses.

/EXPIRED

Selects files according to the planned expiration date for each file. This keyword is relevant only with the /BEFORE or /SINCE keywords, and should not be used with the /CREATED or /MODIFIED keywords.

/FULL

Lists the following items for each file:

- file name
- file type
- version number
- number of blocks used
- number of blocks allocated
- date of creation
- date last modified
- date of expiration
- file owner's UIC
- file protection
- file identification number (FID)
- file organization
- other file attributes
- record attributes
- record format

You can find descriptions of these items in the Introduction to VAX-11 Record Management Services. Refer to Figure 4-1 at the end of this section for an example of this keyword used with the Directory command.

The /FULL keyword overrides the default brief listing format.

/HEADING (default)
/NOHEADING

Controls whether heading lines consisting of a device description and directory specification are printed. The default output format provides this heading.

/MODIFIED

Selects files according to the last date the file was modified. This keyword is relevant only with the /BEFORE or /SINCE keywords, and should not be used with the /CREATED or /EXPIRED keywords.

/OUTPUT[=file-spec]

Requests that the Directory command output be written to the file specified rather than to the current SYS$OUTPUT device. If you specify the /OUTPUT keyword without a file specification, the output is directed to SYS$OUTPUT. If you omit the file type in the file specification, the default file type is LIS.

Wild card characters are not allowed in the file specification.

/OWNER
/NOOWNER (default)

Controls whether the file's owner is listed. By default, the owner is not listed.

/PRINTER

Queues the command output for printing under the name given by the /OUTPUT keyword. If you specify /PRINTER without the /OUTPUT keyword, the output is directed to a file named DIRECTORY.LIS, which is spooled for printing automatically and then deleted.

/PROTECTION
/NOPROTECTION
(default)

Controls whether the file protection for each file is listed. The default is /NOPROTECTION, which does not list the file protection.

/SINCE[=time]

Specifies that only those files dated after a specified time be printed. You can specify one of the following values:

absolute-time — An absolute date and time. Observe the syntax rules for date and time values specified in Section 2.7.6.

TODAY — The absolute date and time representing the current day, month, and year, at 00:00:00.0 o'clock.

TOMORROW — The absolute date and time representing 24 hours after 00:00:00.0 o'clock today.

YESTERDAY — The absolute date and time representing 24 hours before 00:00:00.0 o'clock today.

This keyword is normally used in conjunction with one of the following keywords: /CREATED, /EXPIRED, or /MODIFIED. If you omit the /SINCE keyword, you will obtain all the files created, regardless of date. However, if you specify /SINCE without a time or date, you will obtain all files created since today began.

/SIZE[=option]
/NOSIZE (default)

Provides the file size in blocks used and/or allocated for each file listed, according to the option you specify. If you omit this keyword, the default is /NOSIZE. However if you specify only /SIZE without an option, the listing provides the file size in blocks used, by default. The options you can specify are:

ALL — Lists both the file size in blocks used and allocated.

ALLOCATION — Lists the file size in blocks allocated.

USED — Lists the file size in blocks used.

/TOTAL

Inhibits the listing of all individual file information and prints only the trailing lines as described under the /TRAILING keyword.

By default, the output format is /BRIEF, which gives this total, but also lists all the file names, file types, and their version numbers.

/TRAILING (default)
/NOTRAILING

Controls whether trailing lines that summarize the following information are output:

- number of files listed
- total number of blocks used per directory
- total number of blocks allocated
- total number of directories and total blocks used and/or allocated in all directories (only if more than one directory is listed)

By default, the output format includes most of this summary information. The /SIZE and /FULL keywords determine more precisely what summary information is included. If you omit /SIZE or /FULL, only the number of files is printed and possibly the total number of directories, if applicable. If you specify /SIZE, the number of blocks is also printed, according to the size option selected (used and/or allocated). If you specify /FULL, the number of files and the number of blocks used and allocated are all given.

/VERSION=n

Causes the latest n versions of each of the files selected to be listed. If you omit the /VERSION keyword, by default the listing includes all versions of each file.

Examples:

```
     >DIRECTORY
```

The Directory command lists all versions of all files in the current default disk and directory in the brief format. The heading identifies the disk and directory, and the trailing line gives the total number of files.

```
     >DIRECTORY/VERSIONS=1/COLUMNS=1 AVERAGE.*
```

The Directory command lists only the highest versions of all files named AVERAGE in the current default directory. The format is brief, but restricted to just one column. Heading and trailing lines are provided.

```
     >DIRECTORY BLOCK%%%
```

The Directory command locates all versions and types of files in the default device and directory whose names begin with the letters BLOCK and end with any three additional characters. The output format is brief, in four columns, with heading and trailing lines.

```
     >DIRECTORY/TOTAL/SIZE=ALL
```

The Directory command outputs only a header and a trailing line that identifies the total number of files and the blocks used and allocated for all versions of all files in the default disk and directory.

```
     >DIRECTORY/EXCLUDE=(AVER.DAT;*,AVER.EXE;*) [*...]AVER
```

The Directory command locates all versions and types of files named AVER in all directories and subdirectories on the default disk. From this list all versions of all files named AVER.DAT and AVER.EXE are excluded prior to listing and totalling.

```
>DIRECTORY-
>_/MODIFIED/SINCE=09-JUL-1979:01:30/SIZE=ALL/OWNER-
>_/PROTECTION/OUTPUT=UPDATE/PRINTER [A*]
```

The Directory command locates all files that have been modified since 1:30 AM on July 9, 1979 and that reside on the default disk in directories whose names begin with the letter A. It formats the output to include all versions, the size used and allocated, the date last modified, the owner, and the protection codes. The output is directed to a file named UPDATE.LIS that is spooled automatically and deleted when done.

```
> DIRECTORY/FULL *.USG

Directory _DB1:[SMITH]❶
                                    ❹  ❺
CHANGES.USG;8❷          Size:        6/6       Created: 17-OCT-1979 14:52❻
                        Owner:     [360,033]❼   Revised: 17-OCT-1979 15:18 (1)❿
                        File ID:  (12244,3,0)❸  Expires: <None specified>
 ❽File protection:      System:RWED, Owner:RWED, Group:RWED, World:RE
   File organization:   Sequential
   File attributes:     Allocation=6, Extend=0
   Record format:       UFC, 2 byte header
   Record attributes:   Carriage return

Total of 1 file, 6/6 blocks.❾
```

Figure 4-1 Sample of Output of Directory Command

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

The following notes describe the sample Directory command listed above in Figure 4-1.

❶ Disk and directory name

❷ File name, file type, and version number of each file

❸ File identification number (FID) in the format;

(file-number, file-sequence-number,relative-volume-number)

❹ Number of blocks occupied by the file

❺ Number of blocks allocated for the file

❻ Date and time the file was created or last modified

❼ User identification code of the file's owner in the format:

[group,member]

❽ Protection code associated with the file, in the format:

[system,owner,group,world]

❾ Summary of file information, in the format:

Total of x files, in-use/allocated blocks.

❿ Date and time that this version of the file was last revised, and the revision number

Notes:

- If PIP format directories are desired you can use PIP instead of the Directory command.
- RSX-11M does not support the Directory command.

DMOUNT

4.14 DMOUNT

The Dmount command releases a volume that was previously specified in a Mount command.

Format:

```
DMOUNT[keywords] device-name[:]
```

keyword /[NO]UNLOAD
/UNIT

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:

/UNIT Specifies, for disk volume sets, that only the volume on the specified device is dismounted. By default, the Dmount command dismounts all volumes in a volume set.

/UNLOAD (default)
/NOUNLOAD 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:

```
> MOUNT/SHARE DBA3:PROJ123 DISK
    .
    .
    .
  DMOUNT 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 Dmount 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 Dmount 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.

- If a volume is mounted /SYSTEM or /GROUP, a Dmount 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 vo!'ume 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 Dmount command.

EXAMINE

4.15 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.

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 format and changes the default radix for numeric literals.

/HEXADECIMAL — Indicates that the data is to be displayed in hexadecimal format and changes the default radix for numeric literals. This is the initial default of the command; refer to the notes following the examples 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 the notes following the examples below.

/OCTAL — Indicates that the data is to be displayed in octal format 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 the image to resume execution.

```
> 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 format. 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 format 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 does not support the Examine command.

- 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 (@)

4.16 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

Specifies 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"""
```

The command interpreter strips the set of quotation marks that enclose the entire string and the outermost pair of double quotation marks within the string. This results in:

```
NEVER SAY "DIE"
```

Note that in each case above, the command interpreter will strip the set of quotation marks that enclose the entire string.

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.

4.17 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. 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, either a warning or a severe error message will be displayed.

If you specify an odd value for a status code, either an information message or no message will be displayed.

Example:

```
     >@LEVEL1
```

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.

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:

- RSX-11M does not support the Exit command.

4.18 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
/[NO]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.

Keywords Applicable to Disk and Tape:

/OWNER_UIC=[g,m]

Specifies the user identification code (UIC) 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 accesss.

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 (OPER) 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 faulty 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.

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 BLOCK LOCATOR Utility (BAD), which is described in the VAX-11 Utilities Reference Manual.

/CLUSTER_SIZE=n

Defines the minimum allocation unit in blocks. The maximum size that you can specify for a volume is 1/100 the size of the volume. The minimum size that you can specify is calculated by using the formula:

```
(disk size)/(255 * 4096)
```

The default cluster size for a Files-11 Structure Level 2 disk depends on the disk capacity. For disks that are 50,000 blocks or larger, the default cluster size is 3. Disks that are smaller than 50,000 blocks have a default value of 1.

Files-11 Structure Level 1 disks always have a cluster size value of 1.

/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.

/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 default 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 user default file protection specified in your user authorization file.

/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 keyword 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.

/MAXIMUM_FILES=n

Restricts the maximum number of files that the volume can contain which can override the default value. The default is calculated from the volume size in blocks as follows:

(volume-size)/((cluster factor +1)* 2)

The maximum size that you can specify for any volume is determined by the following formula:

(volume-size)/(cluster factor +1)

The minimum value that you can specify is 0. However, note that once a value for the maximum number of files is specified, the only way to reset it is to reinitialize the volume.

/SHARE (default)
/NOSHARE

Controls whether a disk volume is shareable. The protection code for the volume defaults to all types of access for all categories of users. If you specify /NOSHARE the protection code defaults to no access for group or 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 from one through 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)
/NOVERIFIED

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.

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 800, 1600, or 6250.

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, separate each by a comma and enclose them in parentheses.

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.

Notes:

- 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.
- The compatible form of the Initialize command between the VAX/VMS and RSX-11M systems is:

```
INITIALIZE device-name:volume-label
```

LIBRARY

4.19 LIBRARY

The Library command replaces a module in an object, macro, help, or text library; creates or modifies libraries; inserts, deletes, extracts, or lists the modules or symbols within a library.

Libraries are files that contain one or more directories pointing to the locations of individual modules. The Library command creates libraries and modifies their contents. You use VAX/VMS MCR or DCL commands to manipulate a library in its entirety. For example, the Delete, Copy and Rename commands delete, make copies of, or rename libraries, respectively.

The Library command distinguishes four types of libraries:

- Object module libraries contain frequently called routines. You can use object module libraries as input to the linker. The linker searches the object module library whenever it encounters a reference it cannot resolve from the specified input files.

- Macro libraries contain macro definitions. You can use macro libraries as input to the assembler. The assembler searches the macro library whenever it encounters a macro that is not defined in the input file.

- Help libraries contain help text. You can retrieve help messages by calling the appropriate library procedures from your program. See the VAX-11 Utilities Reference Manual for information about calling library procedures.

- Text libraries contain any sequential record file that you want to retrieve as data for your program. You can retrieve text from text libraries by calling the appropriate library procedures from your program. See the VAX-11 Utilities Reference Manual for information about calling library procedures.

All libraries contain a directory called a module name table (MNT) that names the modules in the library. Object module libraries also contain a global symbol table (GST) that is a list of the global symbols defined in each of the modules in the library. When the Library command adds a module to a library, it catalogs the module by its module name, rather than the input file specification. The only exception to this procedure occurs with text libraries, where the name of the input file containing the text automatically becomes the module name.

When using the Library command, you can specify keywords that request more than one function in a single command, with some restrictions. Generally, you cannot specify multiple keywords that request incompatible functions. The keywords that perform library functions, related keywords, and keyword incompatibilities are summarized in Table 4-1.

Table 4-1
Library Command Keywords

Keyword	Related Keywords	Incompatible Keywords
/COMPRESS	/OUTPUT	/CREATE, /EXTRACT
/CREATE[1]	/SQUEEZE[2] , /GLOBALS[3] , /SELECTIVE_SEARCH[3]	/COMPRESS, /EXTRACT
/CROSS_REFERENCE	/ONLY	/EXTRACT
/DELETE	---	/EXTRACT
/EXTRACT	/OUTPUT	/COMPRESS, /CREATE, /DELETE /INSERT, /LIST, /REMOVE, /REPLACE
/INSERT	/SQUEEZE[2] , /GLOBALS[3] , /SELECTIVE_SEARCH[3]	/EXTRACT
/LIST	/FULL, /NAMES[3] , /ONLY	/EXTRACT
/MODULE[4]	/TEXT	/EXTRACT, /DELETE, /REMOVE
/REMOVE[3]	---	/EXTRACT
/REPLACE	/SQUEEZE[2] , /GLOBALS[3] , /SELECTIVE_SEARCH[3]	/EXTRACT

1. The /CREATE, /INSERT, and /REPLACE keywords are not incompatible; however, if you specify more than one, then /CREATE takes precedence over /INSERT, and INSERT takes precedence over /REPLACE. The related keywords for /CREATE are applicable only if you enter one or more input files.

2. Indicates a keyword that applies only to macro libraries

3. Indicates a keyword that applies only to object libraries

4. Indicates a keyword that applies only to text libraries

For more information on VAX/VMS libraries, see the VAX-11 Utilities Reference Manual.

Format:

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

keywords            /COMPRESS[=(option[,...])]
                    /CREATE[=(option[,...])]
                    /CROSS_REFERENCE[=(option[,...])]
                    /DELETE=(module[,...])
                    /EXTRACT=(module[,...])
                    /FULL
                    /[NO]GLOBALS
                    /HELP
                    /INSERT
                    /[NO]LOG
                    /MACRO
                    /OBJECT
```

```
/[NO]NAMES
/ONLY=(module[,...]
/REMOVE=(symbol[,...])
/REPLACE
/SELECTIVE_SEARCH
/[NO]SQUEEZE
/TEXT
/WIDTH=n
/MODULE=module-name
```

output-file-spec

Specifies the name of the output file when the /EXTRACT or /COMPRESS keywords are specified. No wild card characters are permitted in the file specification.

lib-file-spec

Specifies the name of the library you want to create or modify.

Wild card characters are not allowed in the library file specification.

If the file specification does not include a file type, the Library command assumes a default type of OLB, which indicates an object library.

NOTE

Any attempt to modify a library that was created by the VAX/VMS Version 1.0 Librarian, results in an automatic compression into the new format introduced with Version 2.0. The compression occurs prior to the requested modification (See the /COMPRESS keyword at the beginning of the command keyword section.)

Furthermore, libraries created prior to Version 2.0 that have not been modified or compressed appear in a different format when listing files are requested.

input-file-spec[,...]

Specifies the names of one or more files that contain modules you want to insert into the specified library.

Whenever you include an input file specification, the Library command either replaces or inserts the modules contained in the input file(s) into the specified library. The input-file-spec parameter is required when you specify either /REPLACE (the LIBRARY command's default operation) or /INSERT, which is an optional keyword.

When you use the /CREATE keyword to create a new library, the input-file-spec parameter is optional. If you include an input file specification with /CREATE, the Library command first creates a new library, and then inserts the contents of the input file(s) into the library.

Note that the /EXTRACT keyword does not accept an input file specification.

If you specify more than one input file, separate each file specification with a comma (,). The Library command will then insert the contents of each file into the specified library.

If any file specification does not include a file type, the Library command assumes a default file type of OBJ, designating an object library. You can control the default file type by specifying the appropriate keyword as indicated below:

Keyword	**Default File Type**
/HELP	HLP
/MACRO	MAR
/OBJECT	OBJ
/TEXT	TXT

Note also that the file type you specify on the lib-file-spec parameter can affect the default file type of the input file specification, provided the /CREATE keyword is not being issued. For example, if the library file type is HLB, MLB, OLB, or TLB, the input file type default is HLP, MAR, OBJ, or TXT, respectively.

Command Keywords:

/COMPRESS[=(option[,...])]

Requests the Library command to perform either of the following functions:

- Recover unused space in the library resulting from module deletion, or:

- Reformat a library created by the VAX/VMS Version 1.0 Librarian into a Version 2.0 format.

When you specify /COMPRESS, the Library command by default creates a new library with a version number one higher than the existing library. Provide an output file specification to specify an alternate name for the compressed library.

Specify one or more of the following options to increase or decrease the size of the library, overriding the values specified when the library was created:

BLOCKS:n	Specify the number of 512-byte blocks to be allocated for the library
GLOBALS:n	Specify the maximum number of global symbols the library can contain (for object module libraries only)

KEYSIZE:n — Change the maximum length of a module name or global symbol

MODULES:n — Specify the maximum number of modules or macros the library can contain

If you specify more than one option, separate each with a comma and enclose the list in parentheses.

/CREATE[=(option[,...])]

Requests the Library command to create a new library. When you specify /CREATE, you can optionally specify a file or a list of files that contains modules to be placed in the library.

By default, the Library command creates an object module library; specify /MACRO, /HELP, or /TEXT to change the default library type.

Specify one or more of the following options to control the size of the library, overriding the system defaults:

BLOCKS:n — Specifies the number of 512-byte blocks to be allocated for the library. By default, the Library command allocates 100 blocks for a new library. The value n is interpreted as a decimal number.

GLOBALS:n — Specifies the maximum number of global symbols the library can contain initially. By default, the Library command sets a maximum of 128 global symbols for an object module library. The value n is interpreted as a decimal number. (Macro, help, and text libraries do not have a global symbol directory; therefore, the maximum for these libraries defaults to 0.)

KEYSIZE:n — Defines the maximum name length of modules and global symbols. By default the Library command limits the names of object, macro, and text modules and global symbols to 31 characters. The limit for help modules is 15 characters. The value n is interpreted as a decimal number.

When you specify a keysize value, remember that VAX-11 MACRO and the linker will not accept module names or global symbol names in excess of 31 characters.

MODULES:n — Specifies the maximum number of modules the library can contain. By default, the Library command sets an initial maximum of 512 modules for an object module library and 256 modules for all other libraries. The value n is interpreted as a decimal number.

An index in a library can grow past its initial allocation. However, for optimum performance, it is best to allocate the maximum number of modules you expect to use.

If you specify more than one option, separate each with a comma and enclose the list in parentheses.

/CROSS_REFERENCE[=(option[,...])]

Requests a cross reference listing of an object library.

If you omit this keyword, cross reference listings are not provided. However, if you specify /CROSS_REFERENCE without specifying an option, you will obtain cross reference listings by default that contain only references to names and values of symbols.

You may specify one or more of the following options:

ALL — Specifies that all types of cross references are desired

MODULE — Specifies a cross reference of both the global symbol references in the module and the global symbol definitions

NONE — Specifies that no cross reference listing is desired

SYMBOL — Provides a cross reference by symbol name

VALUE — Provides a cross reference of symbols by value

If you specify more than one option, separate each with a comma and enclose the list in parentheses.

/DELETE=(module[,...])

Requests the Library command to delete one or more modules from a library. You must specify the names of one or more modules to be deleted from the library. If you specify more than one module, separate each with a comma and enclose the list in parentheses.

Wild card characters are allowed in the module specification. See the chapter on file specifications in the VAX/VMS Command Language User's Guide for more information on wild card characters.

If you specify the /LOG keyword in conjuction with /DELETE, the Library command issues the message:

```
%LIBRAR-S-DELETED, MODULE module-name DELETED FROM
library-name
```

The Library command physically removes modules from a library.

/EXTRACT=(module[,...])

Copies one or more modules from an existing library into a new file. If you specify more than one module, separate each module name with a comma and enclose the list in parentheses.

Wild card characters are allowed in the module specification. See the chapter on file specifications in the VAX/VMS Command Language User's Guide for more information on wild card characters

If you specify an output file specification in conjunction with /EXTRACT, the Library command writes the output into the specified output file. If you specify /EXTRACT and do not specify an output file specification, the Library command writes the output into a file that has the same file name as the library and a file type of OBJ, MAR, HLP, or TXT depending on the type of library.

/FULL

Requests a full description of each module in the module name table.

/GLOBALS (default)
/NOGLOBALS

Control, for object module libraries, whether the names of global symbols in modules being inserted in the library are included in the global symbol table.

By default, the Library command places all global symbol names in the global symbol table. Use /NOGLOBALS when you do not want global symbol names in the global symbol table.

/HELP

Indicates that the library is a help library. When you specify the /HELP keyword, the library file type defaults to HLB and the input file type defaults to HLP.

For information on how to create help files, see the VAX-11 Utilities Reference Manual.

/INSERT

Requests the Library command to add the contents of one or more files to an existing library. If an object module file specified as input consists of concatenated object modules, the Library command creates a separate entry for each object module in the file; each module name table entry reflects an individual module name. If a macro or help file specified as input contains more than one definition, the Library command creates a separate entry for each one, naming the module name table entries according to the names specified on the .MACRO directives or in the HELP format.

When the Library command inserts modules into an existing library, it checks the module name table before inserting each module. If a module name or global symbol name already exists in the library, the command issues an error message and does not add the module to the library.

To insert or replace a module in a library regardless of whether there is a current entry with the same name, use the /REPLACE keyword.

/LOG
/NOLOG (default)

Control whether the Library command verifies each library operation. If you specify /LOG, the Library command displays the module name, followed by the library operation performed, followed by the library file specification.

/MACRO

Indicates that the library is a macro library. When you specify /MACRO, the library file type defaults to MLB and the input file type defaults to MAR.

/NAMES
/NONAMES (default)

Control, when a listing file is specified for a library, whether the Library command lists the names of all global symbols in the global symbol table as well as the module names in the module name table.

The default is /NONAMES, which does not list the global symbol names. If you specify /NAMES, each module entry name is displayed in the format:

```
Module module-name
global-symbol     global-symbol     global-symbol     global-symbol
      .                 .                 .                 .
      .                 .                 .                 .
      .                 .                 .                 .
```

If the library is a macro, help, or text library and you specify /NAMES, no symbol names are displayed.

/OBJECT

Indicates that the library is an object module library. The Library command assumes a library file type of OLB and an input file type of OBJ.

/ONLY=(module[,...])

Specifies the individual modules on which the Library command may operate. When you use the /ONLY keyword, the Library command lists or cross references only those modules specified.

If you specify more than one module, separate each module name with a comma and enclose the list in parentheses.

Wild card characters are allowed in the module name specification(s). See the chapter on file specifications in the VAX/VMS Command Language User's Guide for more information on wild card characters.

REMOVE=(symbol][,...])

Requests the Library command to delete global symbol entries from the global symbol table in an object library. If you specify more than one symbol, separate each with a comma and enclose the list in parentheses.

If you want to verify the names of the deleted global symbols, you must also specify the /LOG keyword.

Wild card characters are allowed in the symbol specifications. See the chapter on file specifications in the VAX/VMS Command Language User's Guide for more information on wild card characters.

/REPLACE (default)

Requests the Library command to replace one or more existing library modules with the modules specified in the input file. The Library command first deletes any existing library modules with the same name as the modules in the input file. Then, the new version of the module is inserted in the library. If any modules contained in the input file do not have a corresponding module in the library, the Library command inserts the new modules in the library.

This is the Library command's default operation. If you specify an input file parameter, the library command either replaces or inserts the contents of the input file into the library. If you use the /LOG keyword with the /REPLACE keyword, the Library command displays, in the following form, the names of each module that it replaces or inserts.

```
%LIBRAR-S-REPLACED, MODULE module-name REPLACED IN
library-file-spec
```

```
%LIBRAR-S-INSERTED, MODULE module-name INSERTED IN
library-file-spec
```

/SELECTIVE_SEARCH

Defines the input files being inserted into a library as candidates for selective searches by the linker. If you specify /SELECTIVE_SEARCH, the linker selectively searches the modules when the library is specified as a linker input file; the linker only includes the global symbol(s) in the module(s) referenced by other modules in the symbol table of the output image file.

/SQUEEZE (default)
/NOSQUEEZE

Control whether the Library command compresses individual macros before adding them to a macro library. When you specify /SQUEEZE, which is the default, trailing blanks, trailing tabs, and comments are deleted from each macro before insertion in the library.

Use /SQUEEZE in conjunction with the /CREATE, /INSERT, and /REPLACE keywords to conserve space in a macro library. If you want to retain the full macro, specify /NOSQUEEZE.

/TEXT

Indicates that the library is a text library. When you use the /TEXT keyword, the library file type defaults to TLB and the input file type defaults to TXT. For more information on text libraries, see the VAX-11 Utilities Reference Manual.

/WIDTH=n

Controls the screen display width (in characters) when listing global symbol names. Specify the /WIDTH keyword with the /NAMES keyword to limit the line length of the /NAMES display.

The default display width is the width of the listing device. The maximum width is 132.

File Keywords:

/MODULE=module-names

Specifies the module name of a text module. Unlike help, object, and macro libraries, text libraries use the file name from the input-file-spec parameter as the module name. If you want the module to have a different name from the input file name, use the /MODULE keyword to identify the added module.

You can also use the /MODULE keyword to enter a text module interactively.

If you specify SYS$INPUT as the input file specification and also issue the /MODULE keyword, the Library command includes the text you enter from the terminal in the specified library module. (To terminate input from the terminal, enter a CTRL/Z.)

Examples

```
>LIBRARY/CREATE = TESTLIB, ERRMSG,STARTUP
```

The Library command creates an object module library named TESTLIB.OLB and places the modules ERRMSG.OBJ and STARTUP.OBJ in the library.

```
>LIBRARY/INSERT = TESTLIB, SCANLINE
>LINK TERMTEST TESTLIB/LIBRARY
```

The Library command adds the module SCANLINE.OBJ to the library TESTLIB.OLB. The library is specified as input to the linker by using the /Library keyword on the Link command. If the module TERMTEST.OBJ refers to any routines or global symbols not defined in TERTEST, the linker will search the global symbol table of library TESTLIB.OLB to resolve the symbols.

```
>LIBRARY/EXTRACT=(ALLOCATE,APPEND) MYHELP = -
SYS$HELP:HELPLIB.HLB
```

The Library command specifies that the modules ALLOCATE and APPEND be extracted from the help library HELPLIB.HLB and output to the file MYHELP.HLP.

```
>LIBRARY/CROSS_REFERENCE=all SYS$OUTPUT = LIBRAR
```

The Library command requests a cross reference listing of the object library LIBRAR.OLB. The cross reference listing is output on the terminal. The listing includes cross references by symbol, by value, and by module.

```
     >LIBRARY/REMOVE=(LIB_EXTRCT_MODS,LIB_INPUT_MAC)/LOG = LIBRAR
```

The Library command requests the removal of the global symbols LIB_EXTRCT_MODS and LIB_INPUT_MAC from the object library LIBRAR.OLB. The /LOG keyword requests that the removal of the symbols be confirmed by messages.

```
     >LIBRARY/MACRO/CREATE=(BLOCKS:40,MODULES:100) = MYMAC, TEMP
     >MACRO MYMAC/LIBRARY,CYGNUS/OBJECT
```

The Library command creates a macro library named MYMAC.MLB from the macros in the file TEMP.MAR. The new library has room for 100 modules in a 40-block file. If the input file contains multiple macros, each macro is entered in the new library.

The Macro command assembles the source file CYGNUS.MAR; the /LIBRARY keyword specifies the library MYMAC.MLB as an input file. If the source file CYGNUS contains any macro calls not defined within the file, the assembler searches the library.

```
     >LIBRARY/INSERT/TEXT = TSTRING, SYS$INPUT/MODULE=TEXT1
```

The Library command inserts a module named TEXT1 into the text library TSTRING.TLB. The input is taken from SYS$INPUT.

Note:

- RSX-11M does not support the Library command.

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

keywords

/BRIEF
/[NO]CONTIGUOUS
/[NO]CROSS_REFERENCE
/[NO]DEBUG[=file-spec]
/FULL
/HEADER
/INCLUDE=module-name[,...]
/LIBRARY
/OPTIONS
/P0IMAGE
/PROTECT
/SELECTIVE_SEARCH
/[NO]SHAREABLE[=file-spec]
/[NO]SYSLIB
/[NO]SYSSHR
/SYSTEM[=base-address]
/[NO]TRACEBACK
/[NO]USERLIBRARY[=(table[,...])]

exe-file-spec

Provides the file specification for the output image file. If you do not provide a file specification, the Link command does not produce an image file. If you omit the file type, the Link command uses EXE as the default file type.

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 file type.

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

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 the file type 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. The /BRIEF keyword 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.

/CONTIGUOUS
/[NO]CONTIGUOUS
(default)

Control whether the output image file is contiguous. The default is /NOCONTIGUOUS.

/CROSS_REFERENCE
/NOCROSS_REFERENCE
(default)

Control 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)

Control whether the image contains the VAX-11 Symbolic 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 keyword 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.

If you specify /DEBUG without a file specification, the default VAX-11 Symbolic Debugger is linked with the image. For information on using the debugger, see the VAX-11 Symbolic Debugger Reference Manual.

/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

/P0IMAGE

Directs the linker to create an image that is stored only in P0 address space. The linker places the stack and RMS buffers that usually go in P1 address space in P0 address space. The /P0IMAGE keyword is used to create executable images that modify P1 address space. See the VAX-11/780 Architecture Handbook for a description of P0 and P1 address space.

/PROTECT

When used in conjunction with the /SHAREABLE keyword, the /PROTECT keyword directs the linker to create a protected shareable image. A protected shareable image can execute privileged change mode instructions even when it is linked into a nonprivileged executable image.

/SHAREABLE
/NOSHAREABLE (default)

Control whether the linker produces a shareable image file rather than an executable image. Shareable images also have a default file type of EXE. The default is /NOSHAREABLE, which means the linker produces an executable file rather than a shareable file, unless the /SHAREABLE keyword is explicitly specified.

/SYSLIB (default)
/NOSYSLIB

Control whether the default shareable image and the system library will be searched for unresolved references. If the linker cannot resolve references with the input file(s), then by default, the linker searches both the default system shareable image, SYS$LIBRARY:VMSRTL.EXE and then the default system library SYS$LIBRARY:STARLET.OLB. If you specify /NOSYSLIB, neither VMSRLT.EXE nor STARLET.OLB is searched.

/SYSSHR (default)
/NOSYSSHR

Control 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 SYS$LIBRARY:VMSRTL.EXE and then the default system library SYS$LIBRARY:STARLET.OLB when it cannot resolve references using the input file(s) specified. Use the /NOSYSSHR keyword 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

Control whether the linker incluaes 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.

/USERLIBRARY[=(table[,...])]
/NOUSERLIBRARY

Control whether the linker searches any user-defined default libraries after it has searched any specified user libraries. When you specify the /USERLIBRARY keyword, the linker searches the process, group and system logical name tables to find the file specifications of the user-defined libraries. (The VAX-11 Linker Reference Manual explains user-defined default libraries.) You can specify the following tables for the linker to search:

ALL — The linker searches the process, group, and system logical name tables for user-defined library definitions.

GROUP — The linker searches the group logical name table for user-defined library definitions.

NONE — The linker does not search any logical name table; this specification is equivalent to /NOUSERLIBRARY.

PROCESS The linker searches the process logical name table for user-defined library definitions.

SYSTEM The linker searches the system logical name table for user-defined library definitions.

If you specify neither /NOUSERLIBRARY nor /USERLIBRARY=(table), the linker assumes /USERLIBRARY=ALL by default.

The /NOUSERLIBRARY keyword tells the linker not to search any user-defined default libraries.

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.

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.

Examples:

```
> LINK MYPROG,MYPROG/FULL MYPROG
```

The above command requests the linking of the object file MYPROG.OBJ. 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

4.21 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:  username [keywords]
Password:  password
```

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

username — 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 2.00
>
```

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 log in 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.

4.22 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]
```

/BRIEF (default) Requests the short form of the logout message. /BRIEF is the default.

/FULL Requests the long form of the logout message.

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 MCR Command Language Interpreter and RSX-11M.

MACRO

4.23 MACRO

Invokes the VAX-11 MACRO assembler to assemble one or more assembly language source programs.

For more information on the VAX-11 MACRO assembler, see the *VAX-11 MACRO User's Guide*.

Format:

```
     MACRO[keywords] [obj-file-spec[keywords]][,lis-file-spec[keywords]]
          =mar-file-spec[keywords][,...]
```

keywords

```
[NO]CROSS_REFERENCE[=function[,...]]
/DISABLE=(function[,...])
/ENABLE=(function[,...])
/LIBRARY
/[NO]SHOW[=(function[,...])]
/UPDATE[=(update-file-spec[,...])]
```

obj-file-spec

Specifies the file specification for the output object file. If you do not provide a file specification, the Macro command does not produce an object module file. If you omit the file type, the Macro command uses OBJ as the file type.

lis-file-spec

Specifies the file specification for the output listing file. If you do not provide a file specification, the Macro command does not produce a listing file. If you omit the file type, the Macro command uses the file type LIS.

Wild card characters are not allowed in the file specification(s).

mar-file-spec[,...]

Specifies one or more VAX-11 MACRO assembly language source files to be assembled. If you do not specify a file type for an input file, the assembler uses the default file type of MAR.

If you specify more than one input file, separate each input file with either a comma (,) or a plus sign (+). The files are concatenated and assembled as a single input file, producing single object and listing files.

Command Keywords:

/CROSS_REFERENCE[=function[,...]]
/NOCROSS_REFERENCE[=function[,...]]
(default)

Control whether a cross reference listing is included in the listing file. The /CROSS_REFERENCE keyword includes a cross reference listing, and therefore requires that a listing file exist. The /NOCROSS_REFERENCE keyword excludes it. You can specify one or more of the functions listed below.

If you specify /CROSS_REFERENCE without any functions, it is equivalent to /CROSS_REFERENCE=(MACROS,SYMBOLS).

ALL	Cross reference directives, macros, operation codes, registers, and symbols
DIRECTIVES	Cross reference directives
MACROS	Cross reference macros
OPCODES	Cross reference operation codes
REGISTERS	Cross reference registers
SYMBOLS	Cross reference symbols

If you specify more than one function, separate each by a comma and enclose the list in parentheses.

/DISABLE=(function[,...])
/ENABLE=(function[,...])

Provide initial settings for functions controlled by the assembler directives .ENABLE and .DISABLE. You must specify at least one of the functions listed below. You can enable or disable:

ABSOLUTE	Assembly of relative addresses as absolute addresses
DEBUG	Inclusion of local symbol table information in the object file for use with the debugger
TRUNCATION	Truncation of floating-point numbers (if truncation is disabled, numbers are rounded)
GLOBAL	Assumption that undefined symbols in the assembly are external symbols
SUPPRESSION	Suppression of the listing of unreferenced symbols in the symbol table
TRACEBACK	Providing information to the debugger traceback mechanism

The default is /ENABLE=(GLOBAL,TRACEBACK).

If you specify more than one function, separate each by a comma and enclose the list in parentheses.

/LIBRARY

Indicates that the associated input file is a macro library. If you do not specify a file type, the assembler uses the default file type of MLB.

If you specify more than one macro library as input files, the libraries are searched in reverse order of their specification when a macro call is issued in a source program.

You must not specify the /LIBRARY and /UPDATE keywords at the same time; they are mutually exclusive.

/SHOW[=(function[,...])]
/NOSHOW[=(function[,...])]

Provide initial settings for the functions controlled by the assembler directives .SHOW and .NOSHOW. You can specify one or more of the functions listed below. If you specify /SHOW without any functions, the listing level count is incremented. If you specify /NOSHOW without any functions, the listing level count is decremented.

The /SHOW keyword requires that a listing file exist.

CONDITIONALS	List unsatisfied conditional code associated with .IF and .ENDC directives
CALLS	List macro calls and repeat range expansions
DEFINITIONS	List macro definitions
EXPANSIONS	List macro expansions
BINARY	List binary code generated by the expansion of macro calls.

If you omit the /SHOW keyword the default is /SHOW=(CONDITIONALS,CALLS,DEFINITIONS).

If you specify more than one function, separate each by a comma and enclose the list in parentheses.

For additional details on these functions, see the VAX-11 MACRO Language Reference Manual.

/UPDATE[=(update-file-spec[,...])]

Indicates that the associated input file is to be updated with the specified update file(s). The batch editor SLP is used. Updating is described in the VAX-11 MACRO User's Guide. The batch editor, SLP, is described in the VAX-11 Utilities Reference Manual.

By default, the assembler uses update files with the same name as the input source file and a file type of UPD.

When multiple update files are specified with the /UPDATE keyword, the assembler merges the contents into a single list of updates before applying the updates to the source file. You must separate multiple update files with commas and enclose the list in parentheses.

As a result of the update the input source file and update file(s) remain unchanged. The effects appear in the compiled output. The listing also provides an audit trail of the changes if you request a listing file with the /UPDATE keyword.

Wild card characters are not allowed in the update file specifications.

You must not specify the /LIBRARY and /UPDATE keywords at the same time; they are mutually exclusive.

If an update file is not found, the assembler prints an information message but continues the assembly.

Examples:

```
> MACRO ORION = ORION
```

The MACRO assembler assembles the file ORION.MAR and creates an object file named ORION.OBJ. If this command is executed in a batch job, the assembler also creates a listing file named ORION.LIS.

```
> MACRO CYGNUS, CYGNUS=CYGNUS
```

This MACRO command assembles CYGNUS.MAR to produce CYGNUS.LIS and CYGNUS.OBJ.

```
> MACRO ALPHA, ALPHA = ALPHA+BETA+MYLIB/LIBRARY+[TEST]OLDLIB/LIBRARY
> PRINT ALPHA.LIS
```

The MACRO command concatenates the files ALPHA.MAR and BETA.MAR to produce an object file named ALPHA.OBJ and a listing file named ALPHA.LIS. MYLIB.MLB (in the current default directory) and OLDLIB.MLB (in the directory TEST) are specified as libraries to be searched for macro definitions. When macro calls are found, OLDLIB, MYLIB, and the system library STARLET.MLB are searched, in that order, for the definitions.

The PRINT command prints the listing file ALPHA.LIS.

Notes:

- RSX-11M does not support this command.
- The entire command name must be typed to invoke the VAX-11 MACRO assembler.

MOUNT

4.24 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)
/BIND=volume-set-name (D)
/BLOCKSIZE=n (T)
/[NO]CACHE[=(option[,...])] (D)
/DATA_CHECK[=option,[...]]
/DENSITY=n (T)
/EXTENSION=n (D)
/FOREIGN
/GROUP (D)
/HDR3 (T)
/[NO]LABEL (T)
/OVERRIDE=(option[,...])
/OWNER_UIC=[g,m]
/PROCESSOR=option
/PROTECTION=code
/[NO]QUOTA (D)
/RECORDSIZE=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.

The volume-label parameter is not required when you mount a volume with the /FOREIGN keyword, nor when you specify /OVERRIDE=IDENTIFICATION. To specify a logical name when you enter either of these keywords, 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.

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.

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. A value from 0 to 255 may be specified to override the default value that was specified when the volume was initialized. The value n is a decimal number.

Operator privilege is required to use /ACCESSED.

/BIND=volume-set-name

Creates or adds to a volume set that consists of one or more disks.

The volume-set-name specifies a 1- to 12 character name that identifies the volume set.

When a volume set is created, the volumes are specified on a volume-label list. Each volume is assigned a relative volume number based on its position on the label list. The first volume specified is called the root volume of the volume set.

When adding one or more volumes to a set, the first volume label specified must be the root volume, unless the root volume already is online.

/CACHE=[(option[,...])]
/NOCACHE[=(option[,...])]

Control whether disk caching limits established at system generation time will be disabled or overridden. One or more of the following limits may be altered using the appropriate option:

[NO]EXTENT[=n]

Enables or disables extent caching. You must have the operator (OPER) privilege and you must specify n, the number of entries in the extent cache. Note that NOEXTENT and EXTENT=0 both disable extent caching.

[NO]FILE_ID=n]

Enables or disables file identification caching denoted by n, the number of entries given to be a value greater than 1. Note that both FILE_ID=1 and NOFILE_ID disable file identification caching. Operator (OPER) privilege is required.

LIMIT=n

Specifies the maximum amount of free space in the extent cache in one thousandths of the currently available free space on the disk.

[NO]QUOTA[=n]

Enables or disables quota cache. Operator (OPER) privilege is required and you must specify n, the number of entries in the cache. Usually, n is set to the maximum number of active users expected on the disk when quotas are enabled. NOQUOTA and QUOTA=0, both disable quota file caching.

WRITETHROUGH

Disables writeback caching, which only writes the file headers of files open for write when other files are closed. Thus, if you specify the WRITETHROUGH option, file headers are written to the disk on every file header operation.

When specifying more than one option, separate each by a comma, and enclose the list in parentheses.

/NOCACHE disables all disk caching on a volume. Specifying /NOCACHE is equivalent to specifying /CACHE=(NOEXTENT,NOFILE_ID, NOQUOTA, WRITETHROUGH).

/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. The value n is interpreted as a decimal number.

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

/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. However, volume protection is 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.

/QUOTA (Default)
/NOQUOTA

Controls whether disk quotas will be enforced on this disk volume. The /QUOTA keyword is the default, which enforces quotas for each user. /NOQUOTA inhibits quota checking. You must either own the volume or have the VOLPRO privilege to use this keyword. Refer to Chapter 3 in the VAX/VMS Command Language User's Guide for information concerning disk quotas.

/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.

/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 the volume files are opened. If /WINDOWS is not specified, the number of pointers defaults to the value specified when the volume was initialized. The value n is interpreted as a decimal number.

Operator privilege is required to use the /WINDOW keyword.

Keywords Applicable Only to Tape:

/BLOCKSIZE=n
/NOBLOCKSIZE

Sets the default block size to a specific number of bytes per block, or 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 (in bpi) at which the tape is written for either foreign or unlabeled tapes. You can specify a density of 800, 1600, or 6250. To specify a specific density, either the /FOREIGN or /NOLABEL keyword must be specified and the first operation performed on the tape must be a write.

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 are recorded.

/HDR3 (default)
/nohdr3

Controls whether ANSI HDR3 labels are written on magnetic tapes. ANSI HDR3 labels are written on tapes by default. For tapes which will be used on systems which do not tolerate HDR3 labels, specify /NOHDR3.

/LABEL (default)
/NOLABEL

Indicates whether the tape contains 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.

/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 are 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.

If you specify more than one keyword, separate each with a comma and enclose the entire list in parentheses.

/RECORDSIZE=n

Specifies the number of bytes in each record. This keyword is normally used with the /FOREIGN or /BLOCKSIZE keywords to read and write fixed-length records on a block-structured device. In this case, the record size must be less than or equal to the default block-size. The block size can range from 20 through 65,532 with RMS or 18 through 65,534 without RMS.

Use the /RECORDSIZE keyword whenever you are mounting tapes without HDR2 labels (such as RTM tapes). This provides RMS with default values for both the maximum record size and the length of the largest record.

Examples:

```
> ASN  DBA1:=DK0:
> @RODIN
  MOUNT DK0:MYVOL
% MOUNT-I-MOUNTED, MYVOL mounted on _DBA1:
      .
      .
      .
```

The Assign command defines DBA1 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 DBA1.

```
> 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 llnn:volume-label
  ```

  llnn — 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.

On Statement

4.25 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

4.25.1 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.

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

4.25.2 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.

4.26 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[keyword[,...]][,...]

keywords              /AFTER=absolute-time
                      /BURST
                      /CHARACTERISTIC=(c[,...])
                      /COPIES=n
                      /[NO]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 card characters 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.

/CHARACTERISTICS=(c[,...])

Specifies one or more characteristics desired for printing the file(s). If you specify more than one characteristic, separate each by a comma and enclose the list in parentheses. For details on specifying characteristics, see the VAX/VMS System Manager's Guide.

/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.

/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)

Request 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

Control 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)

Indicate that the specified file(s) contains 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.

File Keywords:

/BURST
/NOBURST (default)

Indicate 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. /NOBURST is the default for user-specified devices.

/COPIES=nn

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 /COPIES as a command keyword, each file named in the command is printed the specified number of times.

/DELETE
/NODELETE (default)

Control whether the file is to be deleted after printing.

If you specify /DELETE after the Print command all files specified are deleted. Files are deleted after printing by default.

Note that the protection code must allow delete access to the curent UIC.

/FEED (default)
/NOFEED

Control 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.

/FLAG_PAGE
/NOFLAG_PAGE (default)

Indicate whether a flag page is to be printed preceding the file. If you specify /FLAG with a file specification, a separate flag page is printed preceding the associated file. The flag page lists the file name.

If you specify /FLAG with the command verb, a flag page is printed for each file in the job; the flag page lists the job name.

/HEADER
/NOHEADER (default)

Control whether the name of the file is printed at the top of each printed page.

/PAGE_COUNT

Specifies the number of pages of the file to be printed. /PAGE_COUNT 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.

/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:

- 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.
- RSX-11M does not support the Print command.

4.27 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 each 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 Keywords:

/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
/NOLOG — Specifies that the names of files be displayed as they are deleted. /NOLOG is the default, thus file names are not displayed as they are purged.

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 (Image)

4.28 RUN (IMAGE)

The Run (Image) command places an image into execution in the process.

You can truncate the Run command to a single letter, R.

Format:

```
     RUN  file-spec
```

keyword /[NO]DEBUG

file-spec Specifies an executable image to be executed. If you do not specify a file type, the Run command uses the default file type of EXE.

No wild card characters are allowed in the file specification.

Command Keywords:

/DEBUG
/NODEBUG Controls, for native VAX-11 images, whether the image is to be run with the debugger. If the image was linked with the /DEBUG keyword 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.

If the image was not linked with the debugger, you can specify /DEBUG to request the debugger at execution time. However, if /NOTRACEBACK was specified when the image was linked, /DEBUG is invalid.

Examples:

```
     > RUN LIBRA
```

The image LIBRA.EXE starts executing in the process.

```
     > MACRO/ENABLE=DEBUG ORION
     > LINK/DEBUG ORION
     > RUN ORION

          DEBUG Version 2.0          3 DEC 1979
     %DEBUG-I-INITIAL, language is MACRO, scope and module set to:

     DBG>
         .
         .
         .

     > RUN/NODEBUG ORION
```

A program is compiled, linked and run with the debugger. Subsequently, a RUN/NODEBUG command requests that the debugger, which is present in the image, not prompt. If an error occurs while the image executes, the debugger can perform traceback and report on the error.

Note:

- The form of the Run command that is compatible between RSX-11M systems and VAX/VMS systems is:

  ```
  RUN file-spec
  ```

RUN (Process)

4.29 RUN (PROCESS)

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.

When you specify keywords with the Run command, the command creates a separate process to execute a specific 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=quota
                     /[NO]AUTHORIZE
                     /BUFFER_LIMIT=quota
                     /DELAY=delta-time
                     /ERROR=file-spec
                     /FILE_LIMIT=quota
                     /INPUT=file-spec
                     /INTERVAL=delta-time
                     /IO_BUFFERED=quota
                     /IO_DIRECT=quota
                     /MAILBOX=unit
                     /MAXIMUM_WORKING_SET=quota
                     /OUTPUT=file-spec
                     /PAGE_FILE=quota
                     /PRIORITY=n
                     /PRIVILEGES=privilege[,...]
                     /PROCESS_NAME=process-name
                     /QUEUE_LIMIT=quota
                     /[NO]RESOURCE_WAIT
                     /SCHEDULE=absolute-time
                     /[NO]SERVICE_FAILURE
```

```
/SUBPROCESS_LIMIT=quota
/[NO]SWAPPING
/TIME_LIMIT=limit
/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)
/NOACCOUNTING

Controls whether accounting records are to be logged for the created process. By default, all processes are logged in the system accounting file.

You must have the user privilege ACNT to disable accounting.

/AST_LIMIT=quota

Specifies the maximum number of Asynchronous System Traps (ASTs) the created process can have outstanding.

If you do not specify an AST limit quota, the default value established at system generation time is used; the minimum required for any process to execute is 2. A value of 10 is typical.

This quota is nondeductible.

/AUTHORIZE (default)
/NOAUTHORIZE

Controls, when the image to be executed is the system login image (LOGINOUT.EXE), whether login searches the user authorization file to validate a detached process.

By default, the login image checks the user authorization file whenever a detached process is created. Specify /NOAUTHORIZE to create a detached process running under the control of the command interpreter. The process permanent files specified by the /INPUT and /OUTPUT keywords are made available to the command interpreter for input and output.

The user privilege DETACH is required to create a detached process. Any nonspecified attributes of the created process default to the same as those of the current process.

/BUFFER_LIMIT=quota

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 at the end of this section.

If you do not specify a buffered I/O limit quota, the default value at system generation time is used; the minimum amount required for any process to execute is 1024 bytes. This quota is pooled; see the notes at the end of this section.

/DELAY=delta-time

Requests that the created process be placed in hibernation and awakened after a specified time interval has elapsed.

If you specify /INTERVAL with /DELAY, the first wake-up request is scheduled for the delay time specified and all subsequent wake-up requests are scheduled according to the specified time interval.

/ERROR=file-spec

Defines a 1- to 63-alphanumeric 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=quota

Specifies the maximum number of files that a process can have open at one time.

If you do not specify an open file quota for a created process, the system uses the default value established at system generation time. The minimum amount required for a process to execute is 2.

This quota is pooled; see the notes at the end of this section.

/INPUT=file-spec

Defines a 1- to 63-alphanumeric 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 placed in hibernation and awakened at regularly scheduled intervals as specified by the value of delta time.

If you specify /DELAY or /SCHEDULE with /INTERVAL, the first wake-up request occurs at the delay time or scheduled time specified. All subsequent wake-up requests occur at the specified interval time. If neither /DELAY nor /SCHEDULE is specified, the first wake-up request occurs immediately.

If the image to be executed is an RSX-11M image, the /INTERVAL keyword has the effect of the /DELAY keyword. Only one wake-up occurs.

/IO_BUFFERED=quota

Specifies the maximum number of system-buffered I/O operations the created process can have outstanding at a time.

If you do not specify a buffered I/O quota, the default value established at system generation time is used; the minimum required for any process to execute is 2. A value of 6 is typical.

This quota is not deductible; see the notes below.

/IO_DIRECT=quota

Specifies the maximum number of outstanding direct I/O operations permitted for the created process at a time.

If you do not specify a direct I/O quota, the default value established at system generation time is used. The minimum value of n that is required for any process is 2.

This quota is not deductible; see the notes at the end of this section.

/MAILBOX=unit

Specifies the unit number of a mailbox to receive a termination message when the created process is deleted.

Mailbox creation and use and process termination mailboxes are described in the VAX/VMS System Services Reference Manual.

/MAXIMUM_WORKING_SET=quota

Specifies the maximum size to which the image to be executed in the process can increase its working set size.

If you do not specify a working set quota, the system uses the default value established at system generation time. The minimum working set size required for a process to execute is 10 pages.

This quota is not deductible; see the notes at the end of this section.

/OUTPUT=file-spec

Defines a 1- to 63-alphanumeric 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.

/PAGE_FILE=quota

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 quota, the system uses the default value established at system generation time. The minimum number of pages required for a process to execute is one 256-page block.

This quota is pooled; see the notes at the end of this section.

/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 real-time priorities.

You must have the ALTPRI 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[,...]

Defines privileges for the created process. The privilege list consists of one or more of the privileges listed in Appendix A. You must have the SETPRV user privilege to give a process you create any privileges that you do not have.

If you specify multiple privileges, each must be separated by a comma, and the list must be enclosed in parentheses.

As listed, any of the privileges 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.

/PROCESS=process-name

Defines a 1- to 15-alphanumeric character 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.

If you do not specify a process name, the process has a null name, by default.

/QUEUE_LIMIT=quota

Specifies the maximum number of timer queue entries that the created process can have outstanding at any one time. This includes timer requests and scheduled wake-up 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 the default value established at system generation time. A process does not require any timer queue limit to execute.

This quota is pooled; see the notes at the end of this section.

/RESOURCE_WAIT
(default)
/NORESOURCE_WAIT

Enables or disables 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 /NORESOURCE_WAIT, the process receives an error status code when the resource is not available.

/RESOURCE_WAIT is the default.

/SCHEDULE=absolute-time

Requests that the created process be placed in hibernation and awakened at the time of day specified by the value of absolute time.

/SERVICE_FAILURE
/NOSERVICE_FAILURE
(default)

Enables or disables 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 /SERVICE_FAILURE 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.

/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 established at system generation time. A process does not require any subprocess quota to execute.

This quota is pooled; see the notes at the end of this section.

/SWAPPING (default)
/NOSWAPPING

Enables or disables swapping for the created process. By default, a process is swapped from the balance set in physical memory to allow other processes to execute. Swapping is similar in function to RSX-11M checkpointing.

You must have the PSWAPM user privilege to specify /NOSWAPPING for a process that you create. If you specify /NOSWAPPING, the process is not swapped out of the balance set when it is in a wait state.

/TIME_LIMIT=limit

Specifies the maximum amount of CPU time allocated to the created process, in delta time, where the resolution is to ten milliseconds. When the time expires, the process is deleted. The default value is established at system generation time. A CPU time limit of 0 indicates that CPU time is not restricted; this is a typical value.

If you restrict CPU time for a process, specify the time limit according to the rules for specifying delta time values, as given in Section 2.7.6.2.

This quota is deductible.

/UIC=[g,m]

Specifies that the created process is to be a detached process. The [g,m] format defines the user identification code (UIC) for the created process. The values of g and m are numeric strings which respectively represent the group and member numbers of the process. The number is in the range of 0 through 377 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 established at system generation time. The minimum number of pages required for a process to execute is 10 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 the requesting process.

```
      > 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.

- Use the following keywords to assign equivalence names for the logical names SYS$INPUT (TI), SYS$OUTPUT (TI), 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 and a detached process is created, pooled quotas are established which restrict the amount of various system resources available to the process and its descendent process(es). The following resource quotas are pooled when you create a detached process; that is, the values you specify are deducted from your current quotas and given to the detached process:

  ```
  /BUFFER_LIMIT
  /FILE_LIMIT
  /PAGE_FILE
  /QUEUE_LIMIT
  /SUBPROCESS_LIMIT
  ```

  The quota amounts are returned to you when the detached process is deleted.

- The system defines minimum values for each specifiable quota; if you specify a quota 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 quotas 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.

- 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-up 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-up 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]
  ```

Set

4.30 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 4-2 briefly summarizes all the Set options. Note that RSX-11M supports only the /UIC set command option.

Most of the Set options, which are summarized below, are equivalent to those described in detail in the VAX/VMS Command Language User's Guide or the VAX/VMS Operator's Guide. Each summary below references the manual in which a full description of the option can be found.

Table 4-2
Set Command Options

Option	Function
ACCOUNTING[3]	Selectively enables or disables the recording of particular kinds of accounting information
CARD_READER[2]	Defines the translation mode for a card reader
[NO]CONTROL_Y[2]	Controls whether the use of the CTRL/Y function key can interrupt an image
DEFAULT[2]	Establishes a disk and/or directory as the current default
DEVICE[3]	Establishes the spooling and error-logging status on a device
HOST[1]	Establishes a virtual communication link between a terminal and a network node to which the terminal is not directly connected.
LOGINS[2]	Establishes the maximum number of users able to log in to the system
MAGTAPE[2]	Defines the density of a magnetic tape device or rewinds the tape
MESSAGE[2]	Overrides or supplements system messages
[NO]ON[2]	Controls whether the command interpreter checks for an error condition following the execution of a command in a command procedure
PASSWORD[1]	Allows users to change their own passwords

[1] Described in this manual.
[2] Described in the VAX/VMS Command Language User's Guide.
[3] Described in the VAX/VMS Operator's Guide.

(continued on next page)

Table 4-2 (Cont.)
Set Command Options

Option	Function
PRINTER[3]	Defines characteristics of a printer
PROCESS[2]	Defines execution characteristics of a process
PROCESS/PRIORITY[2]	Changes the base priority of a process
PROTECTION[2]	Changes the protection applied to a file or a group of files, restricting or allowing access to the file by different categories of users
PROTECTION/DEFAULT[2]	Establishes the default protection for all files subsequently created during the terminal session or batch job
PROTECTION/DEVICE[2]	Established the protection for a non-file-structured device
QUEUE/ENTRY[2]	Changes the attributes associated with one or more entries in a print or batch job queue
RMS_DEFAULT[2]	Provides default multiblock and multibuffer count values to be used by VAX-11 RMS for file operations
TERMINAL[2]	Defines characteristics of a terminal
TIME[3]	Resets the system time
UIC[1]	Changes the UIC of the current process
[NO]VERIFY[2]	Controls whether the command interpreter displays lines in command procedures as it executes them
WORKING_SET[1]	Establishes a default working set size for images executed in the current process

1. Described in this manual.
2. Described in the VAX/VMS Command Language User's Guide.
3. Described in the VAX/VMS Operator's Guide.

SET HOST

4.30.1 Set Host

The Set Host command connects your terminal (through the current host processor) to another processor, called the remote processor. The remote processor must be a VAX-11.

Use the Set Host command to connect to another VAX-11 processor on a network. (Use the Show Network command described in the VAX/VMS Command Language User's Guide to obtain the names of nodes readable to your node.) Once the connection is made, the remote processor prompts for the user name and password. You must have an account on the remote processor to log in.

Once you have connected to the remote processor and logged in, you can use VAX/VMS MCR or DCL commands just as you would on your local processor. You can even use the Set Host command to connect to another remote processor, and so on.

Use the Logout command to log off the last processor you have logged in on. If you have connected to and logged in on more than one processor, the Logout command leaves you logged in on the next-to-last processor.

For example, if your local node is GALAXY, you can use Set Host STAR to connect to the node STAR; you can then use Set Host ORION to connect (still through GALAXY and STAR) to the node ORION.

If you then use the Logout command, you have logged off (and disconnected from) the processor at node ORION, but your are still logged in on (and connected to) the processor at STAR. A second Logout command logs you off STAR, and disconnects you from it. A third Logout command logs you off the local processor, GALAXY.

Format:

```
SET HOST node-name
```

node-name   Specifies the node name for the remote processor.

Example:

```
> SET HOST STAR
Username:
Password:
```

This Set Host command connects the user terminal to the procesor at the network node named STAR. The remote processor then prompts for user name and password. Use the normal login procedure to log in on the remote processor.

Note:

- RSX-11M does not support the Set Host command option.

4.30.2 **Set Password**

The Set Password command allows users to change their own passwords.

To maintain secrecy, users may need to change their passwords from time to time. The Set Password command offers a means of doing this. However, the system manager can control which users have the right to change their passwords.

Passwords may contain from 1 to 31 characters. The valid characters are:

```
A-Z
a-z
0-9
$  (dollar sign)
_  (underscore)
```

When the old and new passwords are entered, the user input is not echoed (to help ensure secrecy). To protect against typing errors that might not be seen when entering the new password, you must enter the desired new password twice.

If an error occurs, the password remains unchanged.

Passwords that exceed six characters are recommended for improved security.

Format:

```
SET PASSWORD
```

Example:

```
> SET PASSWORD
Old password:
New password:
Verification:
```

In response to the Set Password command, the system requests the old password, then the new password. Then the system asks for the new password again for verification purposes. If the user is authorized to change this account's password, the old password is given correctly, and the new password is given identically twice, the password is changed. Otherwise, an error message appears and the password remains unchanged.

Note:

- RSX-11M does not support the Set Password command option.

SET /UIC

4.30.3 Set /UIC

The Set /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 /UIC command.
- To display your UIC, use the Show Process command described in the VAX/VMS Command Language User's Guide.

4.31 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 briefly summarizes all the Show options. Note that RSX-11M does not support any of the Show command options.

Since the Show command options are equivalent to the options of the DCL Show command, these options are described in the VAX/VMS Command Language User's Guide and only are summarized in Table 4-3 below.

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	Print or batch jobs that have been queued but not yet printed or processed
QUOTA	Displays the current disk quota that is authorized and used by a specific user on a specific disk
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

(continued on next page)

Table 4-3 (Cont.)
SHOW Command Options

Option	Displays
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

4.32 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 that is executing in a subprocess or detached 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.

Note that when an image is interrupted by CTRL/Y, and the Run command is issued to execute another image, the interrupted image is also terminated. However, in this case exit handling routines are allowed to execute before the next image is run.

If you interrupt a command procedure by CTRL/Y and you issue the Stop command, or if the Stop command is executed in a command procedure, all command levels are unstacked and control returns to command level 0.

If you specify a process name or process identification, the Stop command terminates the image currently executing in the specified process and deletes the process. If the process is a batch job process, no notification of deletion occurs; the log file for the batch job does not print.

The Group user privilege is required to stop other processes in the same group. The World user privilege is required to stop any process in the system.

For more information on how to use commands like this one in command procedures, consult the VAX/VMS Guide to Using Command Procedures.

Format:

```
     STOP [keyword] [process-name]
```

keyword	/IDENTIFICATION=process-id
process-name	Specifies 1- through 15-alphanumeric character string name of the process to be deleted. The specified process must have the same group number in its user identification code (UIC) as the current process. If you specify the /IDENTIFICATION keyword the process name is ignored. If you specify neither the process-name parameter nor the /IDENTIFICATION keyword, the image executing in the current process is terminated.

Command Keyword:

/IDENTIFICATION=process-id

Specifies the process identification the system assigned to the process when the process was created. When you create a process with the Run command, the Run command displays the process identification number of the process it creates.

When you specify the process identification, you can omit leading zeros.

Examples:

```
^> RUN MYPROG
^Y
> STOP
```

The Run command begins executing the image MYPROG. Subsequently, CTRL/Y interrupts the execution and the Stop command terminates the image.

```
> @TESTALL
^Y
> STOP
```

The @ (Execute Procedure) command executes the procedure TESTALL.COM. CTRL/Y interrupts the procedure and the Stop command returns control to the VAX/VMS MCR command interpreter.

```
^> RUN/PROCESS_NAME=LIBRA  LIBRA
%RUN-S-PROC_ID, identification of created process is 0013340D
    .
    .
    .

> STOP LIBRA
```

The Run command creates a subprocess named LIBRA to execute the image LIBRA.EXE. Subsequently, the Stop command forces the image to exit and deletes the process.

```
> ON ERROR THEN STOP
    .
    .
    .
```

In a command procedure, the On command establishes a default action when any error occurs as a result of a command or program execution. The Stop command stops all command levels; if the On command is executed in a command procedure that is executed from within another procedure, control does not return to the outer procedure, but to the command interpreter.

```
> @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.

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 described in the VAX/VMS Command Language User's Guide displays the current status of subprocesses.

- RSX-11M does not support the Stop command.

STOP/ABORT

4.33 STOP/ABORT

The Stop/Abort command aborts a job that is currently being printed. The /ABORT keyword is required.

Use this command only to abort the printing of jobs entered in the system output queues, that is, the line printer, or terminal queues.

When you issue the Stop/Abort command, the job currently being printed is terminated, and the next job in the queue is dequeued, provided you have sufficient privileges to do so. You can always abort your own job, and you can even abort jobs of other users in your group if you have the Group user privilege. Otherwise, you need the World or Operator user privileges to abort a job that is not your own.

For more information on how to use commands like this one in command procedures, consult the VAX/VMS Guide to Using Command Procedures.

Format:

```
     STOP/ABORT  printer-name[:]
```

printer-name[:] Specifies the name of the printer queue in which the job was entered.

Example:

```
     > STOP/ABORT LPA1:
```

This command aborts the job currently printing on line printer LPA1.

Note:

- RSX-11M does not support the Stop/Abort command.

4.34 STOP/ENTRY

The Stop/Entry command deletes an entry from a batch queue while it is running. The /ENTRY keyword is required.

Use this command to terminate the execution of a batch job while it is running. This command cannot, however, delete a job while it is waiting to be executed. If you want to delete an entry from a device or batch job queue while the entry is waiting to be executed, use the Delete/Entry command.

Note that you can always stop your own job while it is running. You can even stop execution of jobs of other users in your group if you have the Group user privilege. Otherwise, you need the World or Operator user privileges to stop a running job that is not your own.

Format:

```
STOP/ENTRY=job-number queue-name[:]
```

job-number — Specifies the job number of the job to be deleted from the batch queue.

queue-name[:] — Specifies the name of the queue in which the job was entered.

Example:

```
> STOP/ENTRY=230 SYS$BATCH
```

The Stop/Entry command deletes the job associated with the entry number 230 in the batch queue SYS$BATCH.

Note:

- RSX-11M does not support the Stop/Entry command.

STOP/REQUEUE

4.35 STOP/REQUEUE

Stops the printing of the job currently being printed and places that job at the end of the output queue.

When you requeue a job, that job is placed at the end of the queue with its priority level lowered to 1. The next job in the queue is immediately dequeued for printing.

This command is useful when the line printer runs out of paper while it is printing a job; or when a large job of low priority is currently printing one or more other jobs in the queue must be printed immediately.

Note that you can always requeue yor own job. You can even requeue jobs belonging to other users in your own group if you have the Group user privilege. Otherwise, you need the World or Operator user privileges to requeue a job that is not your own.

For more information on how to use commands like this one in command procedures, consult the VAX/VMS Guide to Using Command Procedures.

Format:

```
     STOP/REQUEUE queue-name[:]
```

queue-name[:]   Specifies the name of the queue to be stopped.

Example:

```
     > STOP/REQUEUE LPB0:
```

This command suspends the current print operation on LPB0:, places the job that was currently printing at the end of the queue, and resumes the print operation.

Note:

- RSX-11M does not support the Stop/Requeue command.

4.36 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.

A file or files queued by the Submit command are considered a job. The system assigns a unique job number to each job in the system. When you submit a batch job, the system displays both the job number it assigned to the job and the name of the batch job queue in which it entered your job. (If you would like to suppress this display, you can equate SYS$PRINT to NL:, the null device.)

Batch Job Output: When you submit command procedures for processing by the Submit command, all output from the command procedure is written to a file called name.LOG where name is the file name of the first command procedure file in the job. (Use the /NAME keyword to give the job a different name.) This file is initially written on your default disk; when the batch job completes, the system queues the file to SYS$PRINT and deletes the file after it has printed.

If multiple procedures are submitted, the job terminates if any procedure exits with an error or fatal error status.

For a description of creating and submitting batch jobs, see the VAX/VMS Guide to Using Command Procedures.

Format:

```
     SUBMIT [keywords] file-spec [keyword][,...]

keywords         /AFTER=absolute-time
                 /CHARACTERISTICS=(c[,...])
                 /CPUTIME=n
                 /[NO]DELETE
                 /[NO]HOLD
                 /[NO]IDENTIFY
                 /NAME=job-name
                 /PARAMETERS=parameters[,...]
                 /PRIORITY=n
                 /QUEUE=queue-name[:]
                 /REMOTE
                 /WSDEFAULT=n
                 /WSQUOTA=n
```

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.

If the file specification contains a network node name, the /REMOTE keyword must be specified.

Full wild card characters are allowed in the file specification. See the section on file specifications in the VAX/VMS Command Language User's Guide for detailed information on wild card characters.

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 in Section 2.6.7.

If the specified time has already passed, the job is queued for immediate processing.

By default, jobs are submitted immediately.

/CHARACTERISTICS=(c[,...])

Specifies one or more characteristics desired for printing the file(s). If you specify more than one characteristic, separate each with a comma and enclose the list in parenthesis. For detailed information on specifying characteristics, refer to the VAX/VMS System Manager's Guide.

/CPUTIME=n

Defines a CPU time limit for the batch job. You may specify a delta time (Section 2.6.7), the value 0, or the words NONE or INFINITE for n.

You cannot request more CPU time than permitted by the base limits or the value in your user authorization file. However, you can use this keyword to override the base queue value established by the system manager or the value in your user authorization file, when you need less CPU time than authorized.

If the CPU time is not restricted by the base queue limits or the value in your user authorization file, you can specify 0 or INFINITE to request an infinite amount of CPU time. Specify NONE when you want the amount of CPU time to default to your user authoriztion file value or the limit specified on the queue.

/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.

/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 described in the VAX/VMS Command Language User's Guide.

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.

The ALTPRI 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.

/QUEUE=queue-name[:]

Specifies the name of a specific batch job queue to which the job is to be submitted.

/REMOTE

Indicates that the specified command procedure be executed on a remote node. The file specification must contain the name of the node on which the file resides and at which the procedure is to be executed. See the *DECnet-VAX User's Guide*.

If you specify /REMOTE, you cannot specify any other keywords.

/WSDEFAULT=n

Defines a working set default for the batch job. You may specify a positive integer in the range 1 through 65535, 0, or the word NONE for n.

Use this keyword to override the base queue value established by the system manager or the value in your user authorization file, provided you want to impose a lower value. You may not request a higher value than is specified in your user authorization file.

Specify 0 or NONE if you want the working set value to default to either your user authorization file or the working set default specified on the queue.

/WSQUOTA=n

Defines the maximum size working set for the batch job. This is the working set quota. You may specify a positive integer in the range 1 through 65535, 0, or the word NONE for n.

Use this keyword to override the base queue value established by the system manager or the value in your user authorization file, provided you want to impose a lower value. You may not request a higher value than your default.

Specify 0 or NONE if you want the working set quota to default to either your user authorization file or the working set quota specified on the queue.

File Keyword:

/DELETE
/NODELETE
(default)

Controls whether files are deleted after processing. If you specify the /DELETE keyword after the Submit command name, all files in the job are deleted. If you specify the /DELETE keyword following a file specification, only the associated file is deleted after it is processed.

The protection code on the input file(s) must allow delete access to the default user identification code (UIC) of the user who submitted the job.

Examples:

```
> SUBMIT AVERAGE
  Job 112 entered on queue SYS$BATCH
```

The Submit command enters the procedure AVERAGE.COM in the batch job queue. When the batch job completes, the log file AVERAGE.LOG is queued for printing.

```
> SUBMIT BACKUP/PARAMETERS=(TXT,DOC,MEM), -
>_AVERAGE, RUNMASTER
  Job 416 entered on queue SYS$BATCH
```

The Submit command enters three command procedures in a single job. The job is given three parameters: P1 is equated to the string TXT, P2 to the string DOC and P3 to the string MEM. After the procedure BACKUP.COM is executed, the procedures AVERAGE.COM and RUNMASTER.COM are executed.

```
> BATCH/NAME=BATCH_24/HOLD TESTALL
  Job 467 entered on queue SYS$BATCH
```

The Submit command enters the procedure TESTALL.COM for processing as a batch job, but in a HOLD status. The job will not be released until the Set Queue/RELEASE command is issued. The /NAME keyword requests that the batch job be identified as BATCH_24.

```
>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 described in the VAX/VMS Command Language User's Guide to determine the job identification of a queued batch job.

- The attributes of a queued job can be changed using the Set Queue/ENTRY command as described in the VAX/VMS Command Language User's Guide.

- You can use the Delete/Entry command described in Section 4.11 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

4.37 TIME

The Time command displays the current date and time. This command is equivalent to the Show Daytime and Show Time commands described in the VAX/VMS Command Language User's Guide.

Format:

```
TIME
```

Example:

```
> TIME
  22-FEB-1978  12:43:12
```

Note:

- RSX-11M supports the Time command.

4.38 TYPE

The Type command displays the contents of a file or group of files on the terminal.

Format:

```
     TYPE output-file-spec= input-file-spec,...
```

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 is displayed again when the Type command begins the next file.
- To stop the display completely, press CTRL/Y and then use the Stop command.
- RSX-11M does not support the Type command.

UFD

4.39 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.

/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.

Notes:

- VAX/VMS does not support the /UF switch for PIP to create directories.
- RSX-11M supports the UFD command.

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.

Your command interpreter must be MCR if you want to execute indirect command files. The login procedure described in Section 3.1.1, "Selecting a Command Interpreter," explains how to establish MCR as your command interpreter.

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
	Closes all data and command files and exits	None
.label:	Defines a label	None
!text or ;text	Provides commentary	None
.ASK ssssss txt-string	Asks a question and waits for a reply	None
.ASKN ssssss txt-string	Asks for definition of a numeric symbol	Can specify hexadecimal values; see Section 5.7
.ASKN[low:high] ssssss txt-string[1]	Asks for definition of a numeric symbol specifying an allowable range	Can specify hexadecimal values; see Section 5.7
.ASKN[::def] ssssss txt-string[1]	Asks for definition of a numeric symbol specifying the default value	Can specify hexadecimal values; see Section 5.7
.ASKN[low:high:def] ssssss txt-string[1]	Asks for definition of a numeric symbol specifying an allowable range and the default value	Can specify hexadecimal value; see Section 5.7
.ASKS ssssss txt-string	Asks for definition of a string symbol	None
.ASKS[low:high] ssssss txt-string[1]	Asks for definition of a string symbol specifying the allowable number of characters	None
.BEGIN	Marks the beginning of a Begin-End block	None
.CHAIN file-spec	Continues 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]	Closes secondary file	None
.DATA [#n] txt-string	Writes data to secondary file	None
.DEC ssssss	Decreases value of numeric symbol by 1	None
.DELAY nnu	Delays indirect command file execution for the specified period of time	VAX-11 hardware provides a 100 ticks-per-second clock; under VAX/VMS, hours cannot be specified
.DISABLE DATA [#n]	Disables writing of data to secondary file	None
.DISABLE DOLLARS	Disables dollar sign recognition	RSX-11M does not support this directive; see Section 5.11
.DISABLE ESCAPE	Disables use of escape character as response to .ASK, .ASKN, or .ASKS	None
.DISABLE GLOBAL	Disables definition of global symbols	None
.DISABLE LOWERCASE	Enables lowercase-to-uppercase conversion	None
.DISABLE QUIET	Suppresses echoing of CLI command lines SSSSSSSSS	Ignored by VAX/VMS; see Section 5.5
.DISABLE SUBSTITUTION	Disables symbol substitution	None
.ENABLE DATA [#n]	Enables writing of data to a secondary file	None
.ENABLE DOLLARS	Enables dollar sign recognition	RSX-11M does not support this directive; see Section 5.11

(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	Enables use of escape character as response to .ASK, .ASKN, or .ASKS	None
.ENABLE GLOBAL	Enables definition of global symbols ($sssss)	None
.ENABLE LOWERCASE	Enables use of lowercase characters in strings	Embedded multiple tabs and spaces are not compressed
.ENABLE QUIET	Enables echoing of CLI command lines	Ignored by VAX/VMS; see Section 5.5
.ENABLE SUBSTITUTION	Enables symbol substitution	None
.END	Marks the end of a Begin-End block	None
.ERASE GLOBAL	Erases global symbol definitions	Does not erase permanent global symbols
.ERASE LOCAL	Erases local symbol definitions	None
.EXIT [n]	Terminates processing of a Begin-End block or current command file and optionally sets the special symbol <EXSTAT> value	None
.GOSUB label	Calls a subroutine	None
.GOTO label	Branches to a label	None
.IF ssssss relop expr	Determines whether symbols meet specified conditions	None

(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]ACT tttttt	Determines whether a task is active	Response always indicates that the task is inactive; see Section 5.2
.IF[N]DF ssssss	Determines whether a symbol is defined	None
.IFF ssssss	Determines whether a symbol is false	None
.IF[N]INS tttttt	Determines whether a task is installed	Response always indicates that the task is already installed; see Section 5.2
.IF[N]LOA dd:	Determines whether a loadable driver is loaded	Response always indicates that the driver is loaded; see Section 5.2
.IFT ssssss	Determines whether a symbol is true	None
.INC ssssss	Increases the value of a numeric symbol by 1	None
.ONERR label	Branches to label on detecting an error	Equivalent to an ON WARNING THEN .GOTO label command; see Section 5.3
.OPEN [#n] file-spec	Opens a secondary file for output	None
.OPENA [#n] file-spec	Opens an existing file and appends subsequent data	None
.PAUSE	Pauses for user action	Use the Continue command to resume indirect command file processing

(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
.RETURN	Returns from subroutine	None
.SETF ssssss	Sets the value of a symbol to false	None
.SETN ssssss numexp	Sets a symbol to a numeric value	Can specify hexadecimal value; see Section 5.7
.SETS ssssss strexp	Sets a symbol to a string value	None
.SETT ssssss	Sets the value of a symbol to true	None
.STOP [n]	Terminates indirect command file processing and optionally sets the special symbol <EXSTAT>	None
.TEST ssssss	Tests the length of a string symbol	None

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 a specific file or device.

- You can pass parameter values (P1 through P8) to the indirect file.

The Execute Procedure described in Chapter 4 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 usually do not load drivers. Drivers normally are loaded when the system is initialized. If a particular driver is needed, but has not been loaded, ask the system manager for assistance.

5.3 SUPPORT OF .ONERR

The .ONERR label is equivalent to using the ON WARNING THEN .GOTO label command. See the On Severity-level Statement described in Chapter 4 for more information.

5.4 UNSUPPORTED DIRECTIVES

VAX/VMS does not support the following indirect command file processor directives:

- .XQT and .WAIT

- .ONERR

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 Chapter 4, 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 command file specifications:

- /TR (trace) and /NOTR
- /DE (delete) and /NODE
- /MC (pass command 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.

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.

Table 5-2
Special Symbols

Symbol	Meaning	VAX/VMS-Specific Information
<ALPHAN>	True if last string entered as answer to .ASKS or tested with .TEST contains only alphanumeric characters	--
<DATE>	Assigned the current date; format is dd-mmm-yyyy	--
<DEFAUL>	True if answer to last .ASKN was defaulted	--
<ESCAPE>	True if last query was answered with a single escape character	--
<EXSTAT>	Equivalent to $STATUS	--
<LIBUIC>	Nonprivileged task library UIC	Always [1,54]
<MAPPED>	True if the system on which the image is running is mapped and false if it is unmapped	Always true
<MEMSIZ>	Memory size in K words	Always 32K
<RAD50>	True if last string entered as answer to .ASKS or tested with .TEST contains only Radix-50 characters	--
<RSX11D>	True if RSX-11D system	Always false
<STRLEN>	Length of last string entered as answer to .ASKS or tested by .TEST	--
<SYDISK>	Device mnemonic of system device	Always SY
<SYSTEM>	Octal number representing operating system	Always 5
<SYSUIC>	System UIC	Always [1,54]
<SYUNIT>	Unit number of system device	Always 0
<TIME>	Assigned the current time; format is hh:mm:ss	--
<UIC>	Current directory	The current directory is in a valid VAX/VMS format.

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 radixes:

- Octal (default radix)
- Decimal
- Hexadecimal

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.

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 discard the remaining data records and start processing the next command or directive.

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 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 the indirect command file.

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.

The VAX/VMS Command Language User's Guide provides a more detailed explanation of lexical functions; the following paragraphs summarize syntax rules.

The general format of a lexical function is:

```
     'F$function=name([args,...])
```

'F — Indicates that a function name follows. The substitution operator (') is required.

function-name — Specifies the function to be returned. All function names are keywords. You can truncate function names to any unique truncation.

( ) — Encloses function arguments, if any. The parentheses are required for all functions including those that do not accept arguments.

args,... — 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.

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

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.

- You must have logical I/O privilege (LOG_IO) for VMR to perform logical I/O on system files.

- You must have system protection privilege (SYSPRV) which allows you to create files in other directories.

- 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 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 VMR. Section 5.17.2 provides an indirect command file that can be run to make the necessary logical name assignments.

After you complete system generation, bring the system to an RSX-11M or RSX-11S processor and boot it.

For more information on RSX-11S system generation, refer to the RSX-11S System Generation and Installation Guide.

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 access 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, write, execute, and delete protection for all user categories in all directories.

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 by the Execute Procedure in Chapter 4 and Section 5.15, respectively.

```
.ENABLE SUBSTITUTION
.IF P1 EQ "" .ASKS P1 TARGET DEVICE NAME ! Equates P1 to target device
                                         ! name typed by user

.SETS TEMP P1[<STRLEN>:<STRLEN>]         ! Sets L1 equal to the
                                         ! location of the colon(:),
                                         ! if any, if string P1.
                                         ! Otherwise sets L1 equal to
                                         ! length of P1.
.IF TEMP NE ":" .SETS P1 P1+":"          ! 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
ASN 'P1'=SY0:                            ! target disk
ASN 'P1'=LB:
ASN 'P1'=LB0:
ASN MP:=MP0:                             ! Equates MP0 and MP
ASN TK:=TK0
ASN 'P1'[1,54]BIGTKB.TSK=TKB             ! Assigns equivalence names
ASN 'P1'[1,54]VMR.TSK=VMR                ! for TKB and VMR
```

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.

[NO]ACNT Allows/disallows a process to disable accounting messages for created processes

[NO]ALLSPOOL Allows/disallows a process to allocate spooled devices

[NO]ALTPRI Allows/disallows a process to set priority values

[NO]BUGCHK Allows/disallows a process to make bug check error log entries

[NO]BYPASS Allows/disallows a process to bypass UIC protection in accessing files

[NO]CMEXEC Allows/disallows a process to change mode to executive

[NO]CMKRNL Allows/disallows a process to change mode to kernel and change UICs

[NO]DETACH Allows/disallows a process to create detached processes

[NO]DIAGNOSE Allows/disallows a process to issue diagnostic I/O requests

[NO]EXQUOTA Allows/disallows a process to exceed resource quota

[NO]GROUP Allows/disallows a process to control other processes in the same group

[NO]GRPNAM Allows/disallows a process to place names in the group logical name table

[NO]LOG_IO Allows/disallows a process to issue logical I/O requests to a device

[NO]MOUNT Allows/disallows a process to execute a Queue I/O Request system service to mount a device

[NO]NETMBX Allows/disallows a process to create a network device

[NO]OPER Allows/disallows a process to have operator privileges

PRIVILEGES

[NO]PFNMAP	Allows/disallows a process to map to specific physical pages
[NO]PHY_IO	Allows/disallows a process to issue physical I/O requests to a device
[NO]PRMCEB	Allows/disallows a process to create permanent common event flag clusters
[NO]PRMGBL	Allows/disallows a process to create permanent global sections
[NO]PRMMBX	Allows/disallows a process to create permanent mailboxes
[NO]PSWAPM	Allows/disallows a process to alter its swap mode
[NO]SETPRI	Allows/disallows a process to set base priority to a priority higher than alloted.
[NO]SETPRV	Allows/disallows a process to set other process's privileges
[NO]SHMEM	Allows/disallows a process to create and/or delete structures in shared memory
[NO]SYSGBL	Allows/disallows a process to create system global sections
[NO]SYSNAM	Allows/disallows a process to place names in the system logical name table
[NO]SYSPRV	Allows/disallows a process to attach system status for file access
[NO]TMPMBX	Allows/disallows a process to create temporary mailboxes
[NO]VOLPRO	Allows/disallows a process to override volume protection
[NO]WORLD	Allows/disallows a 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	Total outstanding AST operations plus scheduled wake-up requests
BIOLM	Total outstanding buffered I/O operations
BYTLM	Total outstanding buffer space quota
CPUTIME	Default CPU time limit in milliseconds (0 means infinite)
DIOLM	Total outstanding direct I/O limit
FILLM	Total outstanding open file quota
PGFLQUOTA	Total pages in use in the system paging file
PRCLM	Total outstanding subprocess
TQELM	Total outstanding entries in the timer queue plus temporary common events
WSDEFAULT	Default working set size
WSQUOTA	Maximum working set size

INDEX

VAX-11/RSX-11M
User's Guide
AA-D037B-TE

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 Postag
Necessary
if Mailed in
United Stat

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO.33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

BSSG PUBLICATIONS TW/A4
DIGITAL EQUIPMENT CORPORATION
1925 ANDOVER STREET
TEWKSBURY, MASSACHUSETTS 01876

— — — Do Not Tear - Fold Here — — — — — — — — — — — — — — — — — —