

VAX/VMS Release Notes Version 4.4

Order Number: AA-Z107A-TE

April 1986

This document describes new and changed software features, problems and restrictions to software, changes to documentation, and upgrade information for VAX/VMS Version 4.4.

Revision/Update Information: This manual supersedes previous VAX/VMS Release Notes

Operating System and Version: VAX/VMS Version 4.4

Software Version: VAX/VMS Version 4.4

**digital equipment corporation
maynard, massachusetts**

April 1986

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

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

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

Copyright ©1986 by Digital Equipment Corporation

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

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

The following are trademarks of Digital Equipment Corporation:

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

ZK-3030

**HOW TO ORDER ADDITIONAL DOCUMENTATION
DIRECT MAIL ORDERS**

USA & PUERTO RICO*

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

CANADA

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

INTERNATIONAL

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

In Continental USA and Puerto Rico call 800-258-1710.

In New Hampshire, Alaska, and Hawaii call 603-884-6660.

In Canada call 800-267-6215.

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

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

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

Contents

PREFACE

xi

SECTION 1 UPGRADING TO VERSION 4.4

1-1

1.1 INTRODUCTION

1-1

1.2 UPGRADE CONSIDERATIONS

1-1

1.2.1 Cautions and Restrictions _____

1-2

1.2.2 Layered Products Caution _____

1-2

1.2.3 General Notes _____

1-3

1.2.4 Suggestions _____

1-3

1.2.5 Materials Needed _____

1-3

1.3 UPGRADING A SINGLE SYSTEM

1-4

1.3.1 System Upgrade Summary _____

1-4

1.3.2 Pre-upgrade Procedure _____

1-5

1.3.3 Performing the Upgrade _____

1-10

1.3.3.1 Beginning the Upgrade Procedure • 1-10

1.3.3.2 Upgrade Phase 1 • 1-13

1.3.3.3 Upgrade Phase 2 • 1-19

1.3.3.4 Upgrade Phase 3 • 1-20

1.3.3.5 Upgrade Phase 4 • 1-20

1.3.3.6 Upgrade Phase 5 • 1-20

1.3.4 Post-upgrade Procedures _____

1-21

1.3.4.1 Mandatory Post-upgrade Procedures • 1-21

1.3.4.2 Optional Post-upgrade Procedures • 1-22

1.4 UPGRADING A VAXCLUSTER ENVIRONMENT

1-23

1.4.1 Cluster-Specific Considerations _____

1-23

1.4.2 VAX/VMS Version 4.4-Specific Considerations _____

1-24

1.4.3 Rolling Upgrade Procedure _____

1-25

1.4.4 Concurrent Upgrade Procedure _____

1-27

1.4.5 After the Cluster Upgrade _____

1-29

1.4.5.1 Creating Alternate Roots on a Common
System Disk • 1-29

1.4.5.2 Upgrading DECnet • 1-30

Contents

SECTION 2	NEW AND CHANGED FEATURES	2-1
<hr/>		
2.1	GENERAL USER INFORMATION	2-1
2.1.1	Command Line Recall Capability—Expanded	2-1
2.1.2	DCL (Digital Command Language) Commands—New Commands	2-1
2.1.3	File Specifications and Logical Names—Hyphen Permitted	2-2
2.1.4	VAX Text Processing Utility (VAXTPU)—Changes	2-2
2.1.4.1	Recompiling Section Files • 2-2	
2.1.4.2	Default Section File Type Change • 2-2	
2.1.4.3	EVE\$INIT_KEY and EVE\$CLEAR_KEY in EVE • 2-2	
2.1.4.4	Callable Interface • 2-2	
2.1.4.5	GET_INFO (SYSTEM, "timed_message") • 2-2	
<hr/>		
2.2	SYSTEM MANAGER INFORMATION	2-3
2.2.1	Networking—New and Changed Features	2-3
2.2.2	Cluster Node Names—New Feature	2-3
2.2.3	Show Cluster Utility—New Commands and Changed Features	2-4
2.2.4	System Security—New Command and Attribute Features	2-5
2.2.5	BATCH/PRINT Facility—New Features	2-5
2.2.6	VAX 8600 Memory—Additional Error Logging	2-6
2.2.7	Authorize Utility—Changes	2-6
2.2.7.1	/ATTRIBUTES—New Keyword • 2-6	
2.2.7.2	/ACCESS Qualifier—Enhanced • 2-6	
2.2.7.3	/DEFPRIVILEGES and /PRIVILEGES Qualifiers • 2-6	
2.2.7.4	Secondary Passwords—Change • 2-6	
2.2.7.5	AUTOLOGIN Flag—New Feature • 2-7	
2.2.8	VAX Text Processing Utility (VAXTPU)—Changes	2-7
2.2.8.1	VAXTPU Packaging • 2-7	
2.2.8.2	Changing the Editing Interface from EVE to the EDT Keypad Emulator • 2-7	
2.2.8.3	Using Section Files • 2-7	
2.2.9	RMS—Now Supports Full Shared Sequential Files	2-8
2.2.10	Clusters—Limiting Access to Layered Products	2-8
2.2.11	VAX-11/730 Dual-RL02 System—No Longer Supported	2-10
2.2.12	System Generation Utility (SYSGEN)—New Qualifer	2-11
2.2.13	MOUNT/CACHE=TAPE_DATA	2-11
2.2.14	Monitor Utility—New Commands and Changed Features	2-11
2.2.15	AUTOGEN Command Procedure—New Features	2-12
2.2.16	Show Cluster—Arrow Keys Function Changed	2-12

2.3	APPLICATION PROGRAMMER INFORMATION	2-12
2.3.1	Linker Utility—Debugging Permitted for Shareable Images	2-13
2.3.2	Debugger—New Features	2-13
2.3.2.1	Screen Mode Enhancements • 2-13	
2.3.2.2	Other New Features and Changes • 2-13	
2.3.3	ANALYZE/RMS_FILE Utility—New Commands Added	2-14
2.3.4	Error Log Utility—New Features and Changes	2-14
2.3.4.1	Enhancements to the User Interface • 2-14	
2.3.4.2	/EXCLUDE Qualifier Added • 2-14	
2.3.5	SORT/MERGE Utility—Changes	2-15
2.3.5.1	SORT/MERGE Message Symbols Made Universal • 2-15	
2.3.6	Print Symbiont—User Defined I/O Routines Supported	2-15
2.3.7	VAX/VMS System Services—New Features Added	2-15
2.3.7.1	PRINT USING Function—Behavior Change • 2-16	
2.3.7.2	SORT/MERGE—/INCLUDE and /OMIT Qualifiers • 2-16	
2.3.7.3	SORT/MERGE—%D Radix Designation Error • 2-16	
2.3.8	Run-Time Library—New Features Incorporated	2-16
2.3.8.1	New Procedures • 2-17	
2.3.8.2	New Arguments • 2-18	
2.3.8.3	Obsolete Routines • 2-18	
2.3.8.4	Other Changes • 2-18	
2.3.8.5	Screen Management Restriction • 2-20	
2.3.9	VAX RMS—New Features	2-20
2.3.10	Terminal Driver Support—Changes	2-20
2.3.10.1	Sending a Break • 2-20	
2.3.10.2	Preventing Partial Escape Errors • 2-20	
2.3.10.3	SET HOST/DTE Can Generate a Break • 2-21	
2.3.10.4	SET HOST/DTE/DIAL—Problem and Solution • 2-21	
2.3.10.5	Other Changes • 2-21	
2.3.11	Logical Names Associated With Mailboxes and Mounted Volumes—Changes	2-21
2.3.12	VAX BASIC—Support Changes	2-22
2.3.12.1	Error Messages—Modifications and Additions • 2-22	
2.4	SYSTEM PROGRAMMER INFORMATION	2-22
2.4.1	System Dump Analyzer—New and Changed Features	2-23
2.4.2	CSMA/CD Data Link Drivers—New Features	2-24
2.4.3	TS11/RSX-11—Operation Restriction	2-24
2.4.4	DR11-W/DRV11-WA (XADRIVER)—New Support	2-24
2.4.5	CI Port Driver (PADRIVER)—New Image	2-25
2.4.5.1	Supported Microcode • 2-25	
2.4.5.2	SCS Virtual Circuit Timeouts • 2-25	
2.4.5.3	Virtual Circuit Timeout Errors • 2-26	
2.4.5.4	SYSGEN Parameters • 2-26	

Contents

SECTION 3	PROBLEMS, RESTRICTIONS, AND NOTES	3-1
<hr/>		
3.1	GENERAL USER INFORMATION	3-1
3.1.1	VAX/VMS Document Set—Reorganization _____	3-1
3.1.2	Version 4.0 Release Notes Appendixes—Disposition of Material _____	3-1
3.1.3	Mini-Reference—Supersedes Quick-Reference Booklets _____	3-2
3.1.4	Terminal Driver Line Editing—Clarification _____	3-2
3.1.5	VAX/VMS Backup Utility Reference Manual—Guide For New User's Added _____	3-3
3.1.6	Mail Utility—Restriction in Cluster Running Mixed Versions _____	3-3
3.1.7	VAX/VMS Mail Utility Reference Manual—Text Addition _____	3-3
3.1.8	Guide to Using DCL and Command Procedures on VAX/VMS—Documentation Changes _____	3-3
3.1.9	Extended File Names/Types—Caution _____	3-4
3.1.10	Shutdown Notification on Clusters—Note _____	3-5
<hr/>		
3.2	SYSTEM MANAGER INFORMATION	3-5
3.2.1	STAND-ALONE BACKUP—Mandatory Rebuild _____	3-6
3.2.2	Guide to Multiprocessing on VAX/VMS—Setting Up a VAX-11/782 System _____	3-6
3.2.2.1	Building Multiprocessing Console Diskettes • 3-6	
3.2.2.2	Shutting Down the System • 3-14	
3.2.2.3	Booting the VAX-11/782 System • 3-14	
3.2.2.4	Editing SYSTARTUP.COM • 3-15	
3.2.3	Installation Information—VMB Problem and Solution for RX01/TU58 _____	3-15
3.2.4	VAX/VMS Verify Utility Reference Manual—Text Correction _____	3-16
3.2.5	VAX/VMS Developer's Guide to VMSINSTAL—Text Correction _____	3-16
3.2.6	VAX/VMS Install Utility Reference Manual—Additions and Corrections _____	3-16
3.2.6.1	Enhanced LIST/GLOBAL/FULL Command • 3-16	
3.2.6.2	/SUMMARY Qualifier • 3-16	
3.2.6.3	Corrections to Text • 3-16	
3.2.7	VAX/VMS Accounting Utility Reference Manual—Corrections _____	3-17
3.2.8	Mount Utility Reference Manual—Addition to Manual _____	3-17
3.2.9	VAX/VMS DECnet Test Sender/DECnet Test Receiver Utility Reference Manual—Text Changes _____	3-17
3.2.10	VAX/VMS Authorize Utility Reference Manual—Text Corrections _____	3-17
3.2.11	VAX/VMS Authorize Utility Reference Manual—Error Messages _____	3-18

3.2.12	Authorize Utility—Changes and Notes _____	3-28
3.2.12.1	/ACCESS Qualifier • 3-28	
3.2.12.2	/DEFPRIVILEGES and /PRIVILEGES Qualifiers • 3-28	
3.2.12.3	Secondary Passwords • 3-28	
3.2.12.4	AUTOLOGIN Flag • 3-29	
3.2.13	ACCOUNTING Utility—Abbreviated Qualifier Values Restriction _____	3-29
3.2.14	Image Activation, Search Lists, and Known Images—Note _____	3-29
3.2.15	System Generation Utility (SYSGEN)—Notes and Restrictions _____	3-30
3.2.15.1	UDABURSTRATE Parameter • 3-30	
3.2.15.2	SYSGEN Confuses CONSOL.SYS on VAX 11/780 and VAX 11/785 Systems • 3-30	
3.2.16	VAX/VMS System Generation Utility Reference Manual—Text Changes _____	3-30
3.2.17	User-Created Cluster Quorum File Problem _____	3-31
3.2.18	Batch/Print Facility—Notes _____	3-32
3.2.18.1	SET QUEUE/ENTRY Command—Behavior Change • 3-32	
3.2.18.2	Tab Expansion Determined at Start of Queue • 3-32	
3.2.18.3	Generation of Blank Pages When Set-up or Reset Sequence is Specified • 3-32	
3.2.18.4	Device Reset Sequence and Form Feed Interaction • 3-32	
3.2.19	User Environment Test Package (UETP)—Restriction Removed _____	3-33
3.2.20	VAX RPG II Version 2.0—Restriction _____	3-33
3.2.21	UDA50—Restrictions on Booting From Multiple UDA50 Systems _____	3-33
3.2.22	VAX/VMS System Manager's Reference Manual—VAX-11/730 Dual-RL02 Systems _____	3-33
3.2.23	CI Port—Microcode Revision Level Required _____	3-33
3.2.24	Time Limit during System Bootstrapping—Restriction _____	3-33
3.2.25	MSCP Server Timeout for UDA Disks—Problem and Workaround _____	3-34
3.2.26	Operational Considerations in VAXclusters—Problems and Solutions _____	3-34
3.2.27	Tailored Systems and Layered Products—Installation Information _____	3-34
3.2.28	TECO—Requires Compatibility Mode _____	3-36
3.2.29	VAX LISP Version 1.2—Incompatibility with VAX/VMS Version 4.4 _____	3-36
3.2.30	VAX BASEVIEW—BYTLM Quota _____	3-36
3.2.30.1	VAX/VMS INSTALL Option N—Compatibility Problem • 3-37	
3.2.31	Guide to VAX/VMS Performance Management—Additions to Manual _____	3-37
3.2.32	Monitor Utility—Fails in Certain Cases _____	3-38

Contents

3.2.33	Guide to VAX/VMS System Security—Text Changes	3-38
3.2.33.1	Defining Ownership Privileges • 3-38	
3.2.33.2	Establishing and Changing File Ownership • 3-38	
3.2.33.3	Default ACL Protection • 3-38	
3.2.33.4	Example Change • 3-39	
3.2.34	TU81-Plus—Caching Capability Note	3-39
3.2.35	AIE Ethernet Controller—Unsupported by UETP	3-39
3.2.36	ACMS—Restriction	3-39
3.2.37	Microfiche Listings—Applies to Customers Currently Under Service Contract	3-39
3.2.38	TMPJNL and PRMJNL—Removed	3-40
<hr/>		
3.3	APPLICATION PROGRAMMER INFORMATION	3-40
3.3.1	Guide to Programming on VAX/VMS—Change in Focus	3-40
3.3.2	VAX/VMS Linker Reference Manual—Correction	3-40
3.3.3	Run-Time Library Routines Reference Manual—Additions	3-40
3.3.4	Run-Time Library Screen Management Facility—Restriction	3-41
3.3.5	Debugger—Problems and Restrictions	3-41
3.3.5.1	Debugging Shareable Images—Restriction • 3-41	
3.3.5.2	Debugging SMG Programs—Problem • 3-41	
3.3.5.3	Debugger Changes Affecting Compatibility with Earlier Versions • 3-42	
3.3.6	Terminal Driver Support—Problems	3-43
3.3.6.1	SET HOST/DTE/DIAL Command—DMF-32 Problem and Workaround • 3-43	
3.3.6.2	SET HOST/DTE/LOG Command—Log File Problem • 3-43	
3.3.7	VAX/VMS Command Definition Utility Reference Manual—Example Correction	3-43
3.3.8	VAX Text Processing Utility Reference Manual—Documentation Changes	3-44
3.3.8.1	GET_INFO—Restriction • 3-44	
3.3.8.2	VAX BLISS—VAXTPU Example • 3-44	
3.3.9	Run-Time Library Support of VAX BASIC—USEROPEN Problem	3-48
3.3.10	PL/I PRINT FILE Format—Line-Feed Change	3-48
3.3.11	VAX Ada—Compatibility Problem With VAX/VMS Debugger	3-49
<hr/>		
3.4	SYSTEM PROGRAMMER INFORMATION	3-49
3.4.1	DR32 Microcode Loader—Problem Fixed	3-49
3.4.2	VAX/VMS I/O User's Reference Manual: Part II—Additional Information about DR32 Microcode Loader	3-49
3.4.3	VAX MACRO and Instruction Set Reference Manual—Additional Information on Cyclic Redundancy Check (CRC)	3-49
3.4.4	\$PRDEF Symbols—Documentation Addition	3-50

3.4.5	DR32 Bitfield Definitions—Corrections _____	3-51
3.4.6	LPA11-K Driver (LADRIVER)—Changing Timeouts Allowed _____	3-51
3.4.7	CPUDISP Macro—Format Restriction _____	3-51

INDEX

EXAMPLES

3-1	Sample VAX BLISS Template for Callable VAXTPU _____	3-45
-----	---	------

TABLES

2-1	Layered Products with File Names _____	2-9
3-1	Configuration Register Physical Addresses _____	3-8
3-2	Adapter Type Codes _____	3-8

Preface

Intended Audience

This manual is intended for all system users. Please read this manual before you install or use VAX/VMS Version 4.4.

Structure of This Document

This manual supersedes all release notes documentation to previous versions of the VAX/VMS operating system. It includes, or updates, any previous release notes that are still pertinent to the Version 4.4 release.

This manual consists of three major sections:

- Chapter 1 describes the upgrade procedure for the Version 4.4 update.
- Chapter 2 provides a brief summary of each new and changed feature.
- Chapter 3 contains information about problems, restrictions, and notes to the published documentation.

Conventions Used in This Document

The following conventions are observed in this manual:

Convention	Meaning
<code>RET</code>	A symbol with a one- to six-character abbreviation indicates that you press a key on the terminal, for example, <code>RET</code> .
<code>CTRL/x</code>	The phrase <code>CTRL/x</code> indicates that you must press the key labeled <code>CTRL</code> while you simultaneously press another key, for example, <code>CTRL/C</code> , <code>CTRL/Y</code> , <code>CTRL/O</code> .
<code>\$ SHOW TIME</code> <code>05-JUN-1986 11:55:22</code>	Command examples show all output lines or prompting characters that the system prints or displays in black letters. All user-entered commands are shown in red letters.
<code>\$ TYPE MYFILE.DAT</code> . . .	Vertical series of periods, or ellipsis, mean either that not all the data that the system would display in response to the particular command is shown or that not all the data a user would enter is shown.
file-spec,...	Horizontal ellipsis indicates that additional parameters, values, or information can be entered.

Preface

Convention	Meaning
[logical-name]	Square brackets indicate that the enclosed item is optional. (Square brackets are not, however, optional in the syntax of a directory name in a file specification or in the syntax of a substring specification in an assignment statement.)
quotation marks apostrophes	The term quotation marks is used to refer to double quotation marks ("). The term apostrophe (') is used to refer to a single quotation mark.

1 Upgrading to Version 4.4

1.1 Introduction

This chapter describes the VAX/VMS Version 4.4 upgrade procedure. It is structured so that a person who has a minimum of knowledge of the upgrade procedure can perform the upgrade. However, it is assumed in this chapter that the person doing the upgrade does have a knowledge of basic operations on the system that is being upgraded.

This chapter contains information about the following topics:

- Precautions for the upgrade
- Preparations for the upgrade
- Performance of an upgrade on a system
- Performance of an upgrade on a cluster
- Performance of post-upgrade procedures
- References to associated manuals

This chapter is divided into the following sections:

- Section 1.1 – Introduction
- Section 1.2 – Upgrade Considerations
- Section 1.3 – Upgrading a System
- Section 1.4 – Upgrading a Cluster

Before you begin the upgrade procedure, be sure to read Section 1.2, Upgrade Considerations. It contains information that is pertinent to the success of your upgrade.

To begin the upgrade procedure, turn to the section that pertains to your particular configuration, and then follow the numbered steps. You will be provided with all the information you need as you go through the steps and perform the upgrade.

1.2 Upgrade Considerations

This section explains what things you should consider before you begin to upgrade your system.

Upgrading to Version 4.4

1.2.1 Cautions and Restrictions

The following cautions and restrictions **must** be observed for this upgrade:

- VAX/VMS Version 4.4 does not provide support for the upgrade of a VAX 8800 system.
- If the system being upgraded is not currently running VAX/VMS Version 4.2 or 4.3, you **must** upgrade it to at least VAX/VMS Version 4.2 before you begin the Version 4.4 upgrade procedure.
- If current system directories have been altered in any way, the upgrade procedure may not work correctly. You **must** restore your operating system to a standard system before performing the upgrade.
- The upgrade cannot be applied to a tailored system. A tailored system **must** be installed on a separate, initialized disk.
- Upgrades are not supported for RL02, RC25, and RK07 system disks. However, new installations are supported for RC25 and RK07 system disks.
- The system disk and the distribution volume **must not** be moved from one device to another during the upgrade.

1.2.2 Layered Products Caution

The upgrade procedure is engineered so that most layered products should not have to be re-installed after the upgrade. However, certain layered products **must** be re-installed due to product-specific installation procedures. For example, products that create directories that are synonymous with system directories must be re-installed.

In addition, there are some layered products that *recommend* re-installation after an upgrade. To find out if it is recommended that you reinstall your layered product, consult the applicable layered product installation guides for information.

Note the information in the following tables before upgrading your system to VAX/VMS Version 4.4.

The following products must be re-installed after the upgrade:

Product	SPD Number
VAX-11 RSX	26.73.xx
Professional Host Tool Kit	30.28.xx
VAX ENCRYPTION	26.74.xx
VAX RPG II	26.05.xx
DECnet-VAX	25.03.xx

In addition, the current release of some layered products are incompatible with VAX/VMS Version 4.4. Refer to the VAX/VMS System Software Order Table (28.98.xx) for the supported versions of these products.

The releases of the products in the following table are incompatible with VAX/VMS Version 4.4:

Product	SPD Number	Version Number
VAX ACMS Product Set	25.50.xx	V1.2
VAX LISP/VMS	25.82.xx	V1.2
VAX DECReporter	27.10.xx	V1.0
VAX PSI and PSI ACCESS	25.40.xx	V3.2
VAX DEC/Shell	26.69.xx	V1.1

1.2.3 General Notes

The following factors should be noted before installing the upgrade:

- If the upgrade is interrupted for any reason, it can be resumed from the point at which the system was most recently booted. The upgrade reboots the system several times during the upgrade procedure.
- As a part of the upgrade, the paging, swapping, dump, and authorization files are purged to one version.
- Everything in the [SYSERR] directory is deleted.
- All operator and accounting logs are deleted. To save such files, move them to a user directory before starting the upgrade.

1.2.4 Suggestions

The following list contains suggestions that will help in performing the upgrade:

- DIGITAL recommends backing up the system disk before performing the upgrade. If your system is a VAX 8600 or 8650, DIGITAL also recommends backing up the console RL02.
- A major part of the upgrade is automated and does not require the presence of an operator throughout the procedure. For this reason, the upgrade should be performed from a hard-copy device to record all returned data.
- Consider having a second terminal logged in for doing support tasks during the upgrade.

1.2.5 Materials Needed

The following list of materials is required for the VAX/VMS Version 4.4 upgrade:

- A Version 4.4 software distribution kit
- A blank disk for building the upgraded system
- A blank console volume of one of the following types:
 - 1 RX01 floppy diskette for VAX-11/780, VAX-11/782, or VAX-11/785
 - 2 RL02 for VAX 8600 and VAX 8650

Upgrading to Version 4.4

- 3 RX50 floppy diskette for VAX 8200 and VAX 8300
- 4 TU58 cartridge for all other VAX processors

1.3 Upgrading a Single System

This section explains how to upgrade a **single** VAX/VMS system. It is divided into four major sections:

- 1 System Upgrade Summary
- 2 Pre-upgrade Procedure
- 3 Upgrade Procedure
- 4 Post-upgrade Procedure

1.3.1 System Upgrade Summary

Upgrades are installed on existing operating systems to bring them up to the latest major revision from the most recent maintenance version. For example, if you want to bring a Version 4.3 system up to Version 4.4, you would install the Version 4.4 upgrade kit. For the upgrade to be successful, the current system files must be substantially the same as when the system was installed.

A system upgrade is performed in nine steps:

- 1 Back up your current system disk to the disk that will be used for the upgrade.
- 2 Boot the new system volume and log in to the system manager's account (SYSTEM).
- 3 If you want to keep any files that may be affected by the upgrade, move them to a user directory.
- 4 Be sure you have enough space on the new system volume to do the upgrade.
- 5 Be sure the CPU is set up for automatic restart.
- 6 Prevent users from logging into the system, stop all queues, and, if applicable, shut down the network.
- 7 Invoke the VMSINSTAL command procedure and follow instructions that are displayed through the five upgrade phases.
- 8 If any of the system reboots fail, change system parameters as directed.
- 9 When the upgrade is completed, boot the new system and make any required changes in system procedures before resuming normal operations.

1.3.2 Pre-upgrade Procedure

This section prepares the system for the upgrade procedure. Perform the operations in this section before you begin the upgrade procedure. When you have finished this section, go on to the next section. The pre-upgrade procedure is divided into numbered tasks. Each task consists of numbered steps. Follow the steps in the indicated order.

Use the following procedure to prepare for the upgrade:

1. BACKING UP THE SYSTEM DISK

It is recommended that you back up the system disk. The upgrade may fail if this is not done. By backing up the system disk, you accomplish the following:

- The original system disk is preserved.
- Disk performance is improved because all free space on the disk is made contiguous.

To back up the system disk, proceed as follows:

- a Use standalone BACKUP to back up the current system disk. A step-by-step procedure for backing the system is given in the *Guide to VAX/VMS Software Installation* and in the *Guide to VAX/VMS System Management and Daily Operations*.
- b Perform one of the following options, depending upon your system's configuration:
 - If an extra disk drive with a disk of equal capacity is available, you can perform a disk-to-disk backup directly to it from the original system disk. You can then save the original system disk and use the new backup system disk for the upgrade.

To use the new backup disk for the upgrade, you must swap the unit address plugs from the disk drive with the original system disk to the disk drive with the new backup disk. You must do this so that you will be able to boot from the new backup disk.

- If an extra disk drive of equal capacity is not available, you can do a backup from the original system disk to whatever other type of device is available. In this case, you must subsequently perform either of the following options:
 - 1 If the original system disk can be removed, you should remove it and replace it with a scratch disk. Then, reverse the backup process and do a backup from your backup device to the scratch disk. The scratch disk is now your new backup system disk. After you have finished, save the original system disk and use the new backup system disk for the upgrade.
 - 2 If the original system disk cannot be removed, you must use it for the upgrade. However, even though you are using the original system disk for the upgrade, you must still perform a backup operation from the backup device to the original system disk. After you have finished, save the backup media and use the original system disk for the upgrade.

Upgrading to Version 4.4

Note: If you have a VAX 8600 or 8650, you need to back up the console media at this point. Use the command procedure `SY$UPDATE:CONSCOPY` to create an RL02 backup copy of the console media. Refer to the *Guide to VAX/VMS System Management and Daily Operations* for further information about backing up the console media.

2. BOOTING THE BACKUP SYSTEM DISK

Booting is performed in console mode. The system indicates it is in console mode by displaying a `> > >` prompt on the console terminal. Upon completion of the boot, the console program automatically returns the system to program mode. If you have any problems booting the system, refer to the *Guide to VAX/VMS Software Installation* for detailed information.

Boot the backup copy of the system disk (or the restored system disk, if it couldn't be removed) as follows:

- a Put the system into console mode by pressing the CTRL/P keys.
- b Halt the processor by entering HALT and pressing the RETURN key.
- c Invoke the default boot procedure by entering the following command on the console terminal keyboard:

```
>>>B
```

3. CHECKING FREE BLOCKS

A minimum of 20,000 free blocks must be available on the system disk being upgraded. If the space is not available, you must make room as necessary.

- a Log in under the system manager account, SYSTEM, after the new system disk is booted.
- b Confirm the free block count using the following command at the DCL level:

```
‡ SHOW DEVICE ddcu:
```
- c If necessary, create free blocks by deleting or purging files until the minimum of 20,000 is reached.

4. CHECKING AND SETTING PAGE FILE SIZE

The system page file must contain at least 4600 blocks. Check and modify the page file size on your system with the following procedure:

- a Confirm the number of blocks in the system page file by entering the following DCL command:

```
‡ @SYS$UPDATE:SWAPFILES
```

The SWAPFILES procedure will display current file sizes and prompt you to enter new values.

```
Enter new size for paging file:
```

- b Perform one of the following steps:
 - Press the RETURN key to retain the current page file value if it is greater than 4600 blocks.

- Enter the value 4600 and press the RETURN key to modify the page file size if it does not contain at least 4600 blocks.
- c Press RETURN in response to the following two prompts to keep their default values:

```
Enter new size for system dump file:  
Enter new size for swapping file:
```

5. SHUTTING DOWN THE NETWORK

Perform this task only if running DECnet-VAX.

- a Remain logged in under the system manager account, SYSTEM.
- b Shut down the network as follows:

```
‡ RUN SYS$SYSTEM:NCP  
NCP>SET EXECUTOR STATE OFF
```

- c Press CTRL/Z to return to DCL command level.

6. STOPPING QUEUES

Stop all batch and print queues.

- a Determine the state of a queue as follows:
 - ‡ SHOW QUEUE/DEVICE/BATCH/FULL/ALL
- b Bypass the next step if no queues are active.
- c Use the following DCL command to stop each active queue:

```
‡ STOP/QUEUE/NEXT queue_name
```

7. SETTING SYSGEN PARAMETERS

Before beginning the upgrade procedure, the SYSGEN parameters SCSNODE, ALLOCLASS, STARTUP_P1, and VAXCLUSTER must be properly set to the following values:

- VAXCLUSTER must be 0
- ALLOCLASS must be 0
- SCSNODE must be blank
- STARTUP_P1 must be "MIN"
- SCSSYSTEMID must be an unused number

Check and set the values of these parameters as follows:

- a Invoke SYSGEN:
 - ‡ RUN SYS\$SYSTEM:SYSGEN
- b Set for "current":
 - SYSGEN> USE CURRENT

Upgrading to Version 4.4

- c Show and set the VAXCLUSTER parameter to "0":

```
SYSGEN> SHOW VAXCLUSTER
```

Parameter Name	Current	Default	Minimum	Maximum	Unit	Dynamic
VAXCLUSTER	1	1	0	2	Coded-value	

```
SYSGEN> SET VAXCLUSTER 0
```

- d Show and set the SCSNODE parameter to "0":

```
SYSGEN> SHOW SCSNODE
```

Parameter Name	Current	Default	Minimum	Maximum	Unit	Dynamic
SCSNODE	"NODE"	" "	" "	" "	"ZZZZ"	Ascii

```
SYSGEN> SET SCSNODE ""
```

- e Show and set the STARTUP_P1 parameter to "MIN":

```
SYSGEN> SHOW STARTUP_P1
```

Parameter Name	Current	Default	Minimum	Maximum	Unit	Dynamic
STARTUP_P1	" "	" "	" "	" "	"ZZZZ"	Ascii

```
SYSGEN> SET STARTUP_P1 "MIN"
```

- f Show and set the ALLOCLASS parameter to "0":

```
SYSGEN> SHOW ALLOCLASS
```

Parameter Name	Current	Default	Minimum	Maximum	Unit	Dynamic
ALLOCLASS	1	0	0	255	Pure-number	

```
SYSGEN> SET ALLOCLASS 0
```

- g Show and set the SCSSYSTEMID parameter to an unused number (such as 1500):

```
SYSGEN> SHOW SCSSYSTEMID
```

Parameter Name	Current	Default	Minimum	Maximum	Unit	Dynamic
SCSSYSTEMID	1027	0	-1	-1	Pure-number	

```
SYSGEN> SET SCSSYSTEMID 1500
```

- h Save the new parameter settings and exit SYSGEN:

```
SYSGEN> WRITE CURRENT
```

```
SYSGEN> EXIT
```

8. SETTING RESTART SWITCH

- a Make sure the restart switch on the processor control panel is set to automatic restart. This enables the upgrade to continue automatically. Refer to the following chart to determine the proper switch setting for your system.

VAX System	Restart Switch	Required Setting
VAX-11/725	AUTO/RESTART BOOT	ON
VAX-11/730	AUTO/RESTART BOOT	ON
VAX-11/750	POWER ON ACTION	(RESTART)
VAX-11/780	AUTO/RESTART	ON
VAX-11/785	AUTO/RESTART	ON
VAX-8600	RESTART/BOOT	RESTART/BOOT
VAX-8200/8300	AUTO START	ON

9. REBOOTING THE SYSTEM

If any of SYSGEN parameters are modified (as described in Step 6) or if the page file size was changed (as described in Step 4), you **must** reboot the system before continuing the upgrade procedure.

- a** Reboot the system disk as follows:

```
⚡ @SYS$SYSTEM:SHUTDOWN
```

- b** The SHUTDOWN procedure will ask you a series of questions about the system shutdown.

- Answer the questions appropriately.
- Be sure to answer YES when asked if an automatic system boot should be performed.

If you have any problems rebooting the system, or if you want further information about system rebooting, refer to the *Guide to VAX/VMS System Management and Daily Operations*.

10. CONFIGURING SYSTEM DEVICES

Once the system is running, you must reconfigure all available system devices.

- a** Invoke SYSGEN:

```
⚡ RUN SYS$SYSTEM:SYSGEN
```

- b** Configure the system devices:

```
SYSGEN>AUTOCONFIGURE ALL
```

- c** Exit SYSGEN and return to the DCL level:

```
SYSGEN>EXIT
```

- d** Run STARTUP CONFIGURE:

```
⚡ @SYS$SYSTEM:STARTUP CONFIGURE
```

11. ISOLATING THE SYSTEM FROM USERS

- a** Enter the following DCL command to prevent users from logging on the system:

```
⚡ SET LOGINS/INTERACTIVE=0
```

Upgrading to Version 4.4

1.3.3 Performing the Upgrade

This section explains the upgrade procedure for a single VAX/VMS system. It is assumed that you have already completed the pre-upgrade procedure described in the previous section (Section 1.3.1).

The upgrade procedure is divided into the following segments:

- Beginning the upgrade procedure
- Phase 1
- Phase 2
- Phase 3
- Phase 4
- Phase 5

You must perform all parts of the upgrade procedure.

1.3.3.1 Beginning the Upgrade Procedure

During this portion of the upgrade, the VMSINSTAL procedure will ask you a series of questions by displaying prompts. Explanations of the questions and the proper responses are provided in the following steps.

You can enter a question mark (?) for help at any time during the execution of VMSINSTAL. For additional information on invoking VMSINSTAL, see Chapter 5 of the *Guide to VAX/VMS Software Installation*.

Perform the following steps to install the upgrade kit. Respond to upgrade procedure prompts at the operator's console.

1. MOUNTING DISTRIBUTION VOLUME

- a Place the VAX/VMS Version 4.4 distribution volume on the appropriate drive.
 - If you are using a tape drive, thread the tape and put the drive on line. Be sure that the tape is write-protected.
 - If you are using a disk drive, spin the disk up.

2. INVOKING VMSINSTAL

Begin the update procedure by invoking the VMSINSTAL command procedure as follows:

- a Set your default directory to SYS\$UPDATE:

```
‡ SET DEFAULT SYS$UPDATE:
```

- b Enter the VMSINSTAL command:

```
‡ @VMSINSTAL
```

The following VMSINSTAL message is displayed:

```
VMS Software Product Installation Procedure V4.4  
It is (date) at (time).  
Enter a question mark (?) at any time for help.
```

3. BACKING UP DISK

The first prompt asks you the following question about disk backup:

***Are you satisfied with the backup of your system disk [YES]?**

- If you have already backed up the system disk and are satisfied with it, do the following:
 - a Press the RETURN key.
 - b Go on to the next step and continue with the upgrade.
- If you have not backed up the old system disk or are unsatisfied with the previous backup, or if you have a VAX 8600 or 8650 and have not backed up the console RL02, do the following:
 - a Enter *NO* and press the RETURN key. VMSINSTAL returns to the DCL level so the backup can be performed.
 - b Refer to the pre-upgrade section (Section 1.3.1) for instructions about backing up the system and console disks.
 - c Begin the upgrade procedure again from the beginning of this section when the backup operation is completed.

4. SPECIFYING DEVICE NAME

Next, VMSINSTAL requests the name of the drive holding the distribution volume (load device):

***Where will the distribution volumes be mounted:**

- a Enter the device name and press the RETURN key.
 - If the distribution volumes are to be mounted on a device other than an HSC device, use the format **ddcu** when you answer this prompt.
 - If the distribution volumes are to be mounted on an HSC device, use the format **hsc name\$ddcu** when you answer this prompt.

hsc name identifies the HSC device.
dd specifies the type of device (load device).
c refers to the controller number.
u refers to the device unit number.

For example, if your distribution kit (load device) is a magnetic tape on controller A with unit number 0, you would enter *MSA0*.

- Go on to Step 5 (Specifying Product) if you successfully enter the device name and VMSINSTAL does not return an error message.
 - If VMSINSTAL returns an error message, try re-entering the device name. If an error message is returned again, perform the next step (Step b).
- b Correct the error and try again.

An "invalid device" error message is displayed if an incorrect device name, a nonexistent device, or a device that is not connected is specified. The

Upgrading to Version 4.4

VMSINSTAL procedure will keep prompting you for the device name each time.

- In any of these cases, you must do the following:
 - 1 First, check the device name you are using and make sure you are entering the correct name for the device.
 - 2 Next, check the status of the device itself to be sure it is properly connected and set up.
 - 3 Now, go back to Step a (Enter the Device Name) and try again.
- If you get an error message again, you must do the following:
 - 1 Exit the upgrade procedure by entering CONTROL/Y.
 - 2 Use the DCL command SHOW DEVICE to verify device name and status.
 - 3 Begin the upgrade procedure over again from the beginning of this section.

5. SPECIFYING PRODUCT

VMSINSTAL now prompts for the name of the product to be installed:

***Products:**

- a Enter VMS.
- b Press the RETURN key.

6. READYING DISTRIBUTION VOLUME

Finally, you are asked if the distribution volume is ready for mounting on the selected device:

***Are you ready?**

This question is asking you if the distribution media has been placed on the load device (drive) and is ready to be mounted by the upgrade procedure.

- a Prepare the distribution media on the load device (drive).
- b Enter Y when the distribution volume is ready for mounting.

The following message is then displayed:

```
%MOUNT-I-MOUNTED, VMS044      mounted on _ddcu:
The following products will be processed:
  VMS      V4.4
          Beginning installation of VMS 4.4 at <date:hh:mm>
%VMSINSTAL-I-RESTORE, Restoring product save set A...
          VAX/VMS V4.4 Upgrade Procedure
          .
          .
          .
```


7. READING INFORMATIONAL MESSAGES

The upgrade procedure now begins displaying various informational messages that do the following:

- Describe what VMSINSTAL is doing.
- Offer notes, suggestions and restrictions about various facets of the upgrade.
- Keep track of the status of the upgrade.

Read these messages carefully to decide whether or not you need to interrupt the upgrade procedure.

8. INTERRUPTING THE UPGRADE

The system allows interruption of the upgrade procedure at various points. Wait for the prompt that asks you if you want to interrupt:

Do you want to continue? (Y/N):

- Press the Y key to continue and begin Phase 1 of the upgrade.
 - Go to the next section, Upgrade Phase 1, for further information.
 - The upgrade procedure will now ask you a series of questions in the form of prompts.
- Press the N key to interrupt the upgrade.
 - If you do interrupt, read and follow the instructions that were provided by VMSINSTAL.
 - You must invoke VMSINSTAL to resume the upgrade.
 - The point at which the upgrade resumes is determined by your system's configuration.

1.3.3.2 Upgrade Phase 1

Three separate sets of instructions are provided for Phase 1 procedures. You will use only one set, depending upon the type of system you are upgrading. You can determine which instructions you should use by finding your system in the following list and turning to the indicated section.

- Upgrading VAX-11/750 systems—turn to **Upgrade Phase 1A**.
- Upgrading VAX 8200/8300 systems—turn to **Upgrade Phase 1B**.
- Upgrading systems other than VAX-11/750 or VAX-8200/8300 systems—turn to **Upgrade Phase 1C**.

Upgrade Phase 1A—for VAX-11/750

This section describes the first phase of the upgrade for users who are upgrading VAX-11/750 systems.

Note: If the system includes CI750, the upgrade procedure provides the option of creating a common system disk for a VAXcluster.

- 1 To ensure system security, the upgrade procedure requires you to change the passwords for the SYSTEM, SYSTEST, and FIELD accounts before continuing. The system requires passwords of eight or more characters.

Upgrading to Version 4.4

- 2 The upgrade procedure now turns off the disk quotas on the system disk and removes directory entries that point to nonexistent files.

You are provided the option of booting with the TU58 rather than directly from the disk. If using a CI750, answer YES to this question. DIGITAL recommends answering YES in any case to ensure you have a console TU58 with the latest versions of the VMB and BOOT58 programs. If you are booting directly from a local system disk, skip to Step 4.

- 3 The upgraded system is built in system root SYSF so the current system is available if needed. At this point, the upgrade procedure guides you through the building of a new console TU58. Specifically, the procedure will change the default bootstrap command procedure (DEFBOO.COMD) on the new console volume to boot from SYSF.

- a Position the BOOT DEVICE switch on the processor control panel to position A.
- b When prompted, insert the current console TU58 in the console drive. The current console volume is used as a base to build the new console volume, but it is *not* altered by the procedure. The procedure copies the old console volume to a disk directory in Files-11 On-disk Structure format.
- c You are then prompted to set DEFBOO.COMD to boot the backup copy of the system disk (assuming you have not done so previously). Using the form dduBOO.COMD, enter the name of the boot file to copy to DEFBOO.COMD.

For example, if the system disk is in DBA1, enter DB1BOO.COMD. If the system disk is controlled by either an HSC or a UDA, the revised version of CIBOO.COMD or DUABOO.COMD that was built when you first installed VAX/VMS must be used. The selected command procedure must be able to boot the system disk without any operator intervention (for example, registers R0 through R5 must be correctly initialized by the command procedure).

Do not specify a conversational boot command file. The upgrade kit is set with parameters that will boot any system. The procedure builds a conversational boot file (UPGEN.COMD) that boots from SYSF. Use this file for only the situations prescribed by the procedure.

- d When the new backup console media is booted, store the old console volume away for safe keeping and insert a blank volume in the console drive. Do not remove this volume for the rest of the upgrade.

Note: Be sure the RECORD button on the new console TU58 cassette is set to allow writing of the cassette.

- e You now have the opportunity to use the Bad Block Locator Utility (BAD) to check the new console volume before it is initialized. BAD checks the new console volume for any defective blocks. If running BAD, allow an additional 30 minutes for this procedure.

Details on running BAD can be found under the ANALYZE/MEDIA command in the *VAX/VMS DCL Dictionary* or under Bad Block Locator in the *VAX/VMS Bad Block Locator Utility Reference Manual*. DIGITAL suggests performing the BAD check.

- 4 At this point the upgrade procedure cleans up directories on the system disk, removes installed images, and initializes the new console volume. A series of messages is displayed indicating the state of the upgrade.

Upgrading to Version 4.4

- 5 Eventually, the upgrade procedure indicates it will shut down to reboot the partially installed Version 4.4 system. The procedure should reboot automatically. However, if necessary the upgrade system can be restarted manually as follows:

- If booting from the console TU58, reboot the system from the BOOT58 command level. To invoke BOOT58, press CTRL/P to put the system in console mode and enter the following sequence of commands:

```
>>> B/800 DDAO
BOOT58> B
```

- If booting from the disk, use CTRL/P to enter console mode and reboot with the following command:

```
>>> B/F0000000 ddcu
```

- 6 If the system fails to boot in a CI750 environment because of insufficient nonpaged dynamic memory, use the conversational boot command procedure UPGGEN.CMD to increase the NPAGEDYN system parameter; otherwise, proceed to Step 7.

To invoke UPGGEN.CMD when booting from a disk, use CTRL/P to get into console mode, then enter the following command:

```
>>> B/F0000001 ddcu
```

If booting from a TU58, invoke BOOT58 from console mode using the following command:

```
>>> B/800 DDAO
```

At the BOOT58 prompt, enter the following command:

```
BOOT58> @UPGGEN.CMD
```

For either boot method, enter the following commands to SYSGEN:

```
SYSBOOT> SET NPAGEDYN 120000
SYSBOOT> CONTINUE
```

- 7 During the shutdown, the following error message will appear:

```
%SHUTDOWN-I-STOPQUEMAN, the queue manager will now be stopped.
%SYSTEM-F-DEVOFFLINE, device is not in configuration or not available.
```

This error message may be ignored during the upgrade procedure.

- 8 When the system reboots, the following message is displayed:

```
VAX/VMS Version 4.4 <date hh:mm>
```

- 9 This completes Phase 1 of the upgrade. The rest of the upgrade does not require the presence of an operator. However, if booting from the TU58, you will have to manually reboot each time the system shuts down (once in Phase 4 and once in Phase 5). If booting from the disk, reboots are automatic and do not require user intervention.

- 10 Go on to Phase 2.

Upgrading to Version 4.4

Upgrade Phase 1B—for VAX-8200/8300

This section describes the first phase of the upgrade for users who are upgrading VAX-8200/8300 systems.

- 1 To ensure system security, the upgrade procedure requires you to change the passwords for the SYSTEM, SYSTEST, and FIELD accounts before continuing. The system requires passwords of six or more characters.
- 2 The upgrade procedure now turns off the disk quotas on the system disk and removes directory entries that point to nonexistent files.

You are provided the option to boot using the RX50 rather than directly from the disk. If using a CIBCI, answer YES to this question. If booting directly from a local disk, skip to Step 4.

- 3 The upgraded system is built in system root SYSF so the current system is available if needed. At this point, the upgrade procedure guides you through the building of a new console RX50. Specifically, the procedure will change the default bootstrap command procedure (DEFBOO.COMD) on the new console volume to boot from SYSF.
 - a When prompted, insert the current console RX50 in the console drive. The current console volume is used as a base to build the new console volume, but it is *not* altered by the procedure. The procedure copies the old console volume to a disk directory in Files-11 On-disk Structure format.
 - b You are then prompted to set DEFBOO.COMD to boot the backup copy of the system disk (assuming you have not done so previously). Using the form dduBOO.COMD, enter the name of the boot file to copy to DEFBOO.COMD.

If the system disk is controlled by an HSC, the revised version of CIBOO.COMD that was built when you first installed VAX/VMS must be used. The selected command procedure must be able to boot the system disk without any operator intervention (for example, registers R0 through R5 must be correctly initialized by the command procedure).

Do not specify a conversational boot command file. The upgrade kit is set with parameters that will boot any system. The procedure builds a conversational boot file (UPGEN.COMD) that boots from SYSF. Use this file for only the situations prescribed by the procedure.

- c When the backup copy is booted, store the old console volume away for safe keeping and insert a blank volume in the console drive. Do not remove this volume for the rest of the upgrade.
 - d You now have the opportunity to use the Bad Block Locator Utility (BAD) to check the new console volume before it is initialized. BAD checks the new console volume for any defective blocks. If running BAD, allow an additional 30 minutes for this procedure.

Details on running BAD can be found under the ANALYZE/MEDIA command in the *VAX/VMS DCL Dictionary* or under the BAD utility in the *VAX/VMS Bad Block Locator Utility Reference Manual*. DIGITAL suggests performing the BAD check.

- 4 At this point the upgrade procedure initializes the new console volume, cleans up directories on the system disk and removes installed images. A series of messages is displayed indicating the state of the upgrade.

- 5 Eventually, the upgrade procedure indicates it will shut down to reboot the partially installed Version 4.4 system. The procedure should reboot automatically. However, if necessary the upgrade system can be restarted manually as follows:

- If booting from the console RX50, reboot the system from the BOOT58 command level. To invoke BOOT58, press CTRL/P to put the system in console mode and enter the following sequence of commands:

```
>>> B/800 CSA1  
BOOT58> B
```

- If booting from the disk, use CTRL/P to enter console mode and reboot with the following command:

```
>>> B/R5:F0000000 ddn
```

In the above command, the variable n refers to the VAXBI node number.

- 6 If the system fails to boot in a CIBCI environment because of insufficient nonpaged dynamic memory, use the conversational boot command procedure UPGGEN.CMD to increase the NPAGEDYN system parameter; otherwise, proceed to Step 7.

To invoke UPGGEN.CMD when booting from a disk, use CTRL/P to get into console mode, then enter the following command:

```
>>> B/R5:F0000001 ddn
```

In the above command, the variable n refers to the VAXBI node number.

If booting from an RX50, invoke BOOT58 from console mode using the following command:

```
>>> B/800 CSA1
```

At the BOOT58 prompt, enter the following sequence of commands:

```
BOOT58> @UPGGEN.CMD  
SYSBOOT> SET NPAGEDYN 120000  
SYSBOOT> CONTINUE
```

- 7 During the shutdown, the following error message will appear:

```
%SHUTDOWN-I-STOPQUEMAN, the queue manager will now be stopped.  
%SYSTEM-F-DEVOFFLINE, device is not in configuration or not available.
```

This error message may be ignored during the upgrade procedure.

- 8 When the system reboots, the following message is displayed:

```
VAX/VMS Version 4.4 <date hh:mm>
```

- 9 This completes Phase 1 of the upgrade. The rest of the upgrade does not require the presence of an operator. Note that if field service has set the EEPROM so that the default boot device is set to your current system disk, you can set the lower key switch to **Autostart**. In this case, rebooting is automatic and does not require user intervention.
- 10 Go on to Phase 2 of the upgrade.

Upgrading to Version 4.4

Upgrade Phase 1C—for VAX-11/725, VAX-11/730, VAX-11/780, VAX-11/785 and VAX-8600 Systems

- 1 To ensure system security, the upgrade procedure requires you to change the passwords for the SYSTEM, SYSTEST, and FIELD accounts before continuing. The passwords must be at least six characters long.
- 2 The upgrade procedure now turns off the disk quotas on the system disk and removes directory entries that point to nonexistent files.
- 3 The upgraded system is built in system root SYSF. At this point the upgrade procedure will guide you through the building of a new console volume that allows booting the new system from SYSF. Specifically, the procedure will change the default bootstrap command procedure (DEFBOO.COM) on the new console volume to boot from SYSF.

Note: During this section of the upgrade procedure, all command procedures used for upgrading a VAX 8600 (or 8650) will use file extension .COM in place of .CMD.

- 4 The procedure prompts to insert the current console volume in the console drive. The current console volume is used as a base to build the new volume but it is *not* altered by the procedure. The old console volume is copied to a scratch disk directory.
- 5 The procedure then prompts to set DEFBOO.COM to boot the backup copy of the system disk (assuming you have not done so previously). Using the form dduBOO.COM, enter the name of the boot file to copy to DEFBOO.COM.

For example, if the system disk is in DBA1, enter DB1BOO.COM. If the system disk is controlled by either an HSC or a UDA, the revised version of CIBOO.COM or DUABOO.COM built when you first installed VAX/VMS on the HSC- or the UDA-controlled system disk must be used. The selected command procedure must be able to boot the system disk without any operator intervention (for example, registers R0 through R5 must be correctly initialized by the command procedure).

Do not specify a conversational boot command file. The upgrade kit is set with parameters that will boot any system. The procedure does build a conversational boot file (UPGEN.COM) that boots from SYSF, but only use this file for situations prescribed by the procedure.

- 6 When the new console media is complete, store the old console volume for safe keeping and insert a blank volume in the console drive. Do not remove this volume for the rest of the upgrade.

Note: If you have a VAX 8600 or 8650, the old console RL02 is used during this upgrade. A new one is not created; therefore, you should make a backup copy of the console RL02 before performing the upgrade.

If the new console volume is a TU58, be sure the RECORD button on the cassette is set to allow writing of the TU58.

- 7 The procedure now provides the opportunity to have the Bad Block Locator Utility (BAD) check the new console volume before it is initialized. BAD checks the new console volume for any defective blocks. If you choose to run BAD, allow an additional 30 minutes if using a TU58 volume and 15 additional minutes if using a floppy volume.

The use of the BAD utility is recommended. Details on running BAD can be found under the ANALYZE/MEDIA command in the *VAX/VMS DCL Dictionary* or under the description of BAD in the *VAX/VMS Bad Block Locator Utility Reference Manual*.

Note: The Bad Block Locator Utility is not used for upgrades on the VAX 8600.

- 8 The procedure initializes the new console volume and displays various messages describing the state of the build. After several messages are displayed, the upgrade describes the correct method of handling a shutdown or reboot failure. The following information should be noted for these failures:
 - On a VAX-11/730 system, the microcode may be reloaded. If the system fails to boot correctly, power the system off and then on again so that the console will reload the microcode from the newly created TU58.
 - If the system fails to boot in a CI780 environment because of insufficient nonpaged dynamic memory, use the conversational boot command procedure UPGGEN.CMD and increase the NPAGEDYN using the following sequence of commands:

```
>>> @UPGGEN.CMD
SYSBOOT> SET NPAGEDYN 120000
SYSBOOT> CONTINUE
```

When the system reboots, the following is displayed:

```
VAX/VMS Version 4.4 <date hh:mm>
```

During the shutdown, the following error message will appear:

```
%SHUTDOWN-I-ISTOPQUEMAN, the queue manager will now be stopped.
%SYSTEM-F-DEVOFFLINE, device is not in configuration or not available.
```

This error message may be ignored during the upgrade procedure.

- 9 Proceed to Phase 2 of the upgrade.

1.3.3.3 Upgrade Phase 2

The rest of the VAX/VMS Version 4.4 files in the LIBRARY and OPTIONAL save sets are restored from the distribution kit. This step varies, depending on the system configuration, as follows:

- In a standard VAX/VMS configuration with either RK07, RA60, or magnetic tape distribution media, the LIBRARY save set is restored and then the OPTIONAL save set is restored.
- If using RL02 distribution media, the upgrade procedure prompts you to remove the REQUIRED disk and insert the LIBRARY disk. Once the disk is inserted, the LIBRARY save set is restored. The procedure then prompts to remove the LIBRARY disk and insert the OPTIONAL disk. Once inserted, the OPTIONAL save set is restored.

Upgrading to Version 4.4

1.3.3.4 Upgrade Phase 3

Phase 3 of the upgrade merges the VAX/VMS-distributed files that are commonly edited by system managers with new VAX/VMS files. Ignore any "file not found" error messages that appear at this time.

The files HELPLIB.HLB, DCLTABLES.EXE, STARLET.OLB, and IMAGELIB.OLB are modified in an attempt to preserve layered product installations. Modules from the original files are added to the new copies.

Next, the upgrade procedure merges all the miscellaneous user files that exist in the old system directories into a new set of system directories, temporarily called [SYSF.SYSEXEXE], [SYSF.SYSMGR], [SYSF.SYSLIB], and so on. The amount of time the merge takes depends on the number of user files.

1.3.3.5 Upgrade Phase 4

During Phase 4 of the upgrade, the new site-specific console volume is modified to allow reboot of the complete VAX/VMS Version 4.4 system. Be sure the new site-specific console volume created in Phase 1 is in the console drive (CSA1: for VAX-11/780, VAX-11/785, VAX 8600 or 8650 (original RL02), VAX 8200/8300 and VAX-11/750 systems; CSA2: for VAX-11/730 and VAX-11/725 systems).

When the modification is completed, the following message is displayed:

```
System shutting down to boot the complete Version 4.4 system.
```

Leave the newly created site-specific console volume in the console drive. The system disk must also remain where it is for the next phase of the upgrade to proceed. The system will attempt an automatic reboot after the shutdown.

If the system fails to boot because of insufficient nonpaged dynamic memory, you must use a standard conversational bootstrap to increase the NPAGEDYN (nonpaged dynamic memory) system parameter as described in previous sections.

If needed, invoke the conversational boot command procedure for your system disk. For example, if the system disk is on the first MASSBUS device, invoke DBOGEN; if the system is on the first UDA device, DU0GEN, and so forth.

When the boot is completed, the following message is displayed:

```
VAX/VMS Version 4.4 <date hh:mm>
```

1.3.3.6 Upgrade Phase 5

Phase 5 of the upgrade procedure creates a new site-specific SYSGEN parameter file, AUTOGEN.PAR, that combines the default values for Version 4.4 and the site-specific values you were using for the Version 4.2 or 4.3 system.

The upgrade procedure updates the DECnet permanent node database to the new Version 4.4 format by splitting NETNODE.DAT into NETNODE_LOCAL.DAT and NETNODE_REMOTE.DAT.

The upgrade procedure cleans up several directories, announces that the upgrade to Version 4.4 is complete and displays several informational messages that describe various tasks you have the option to perform. Once all the informational messages are displayed, the system shuts down and automatically reboots.

1.3.4 Post-upgrade Procedures

This section describes the mandatory and optional procedures to perform after the upgrade procedure has finished. Refer to the *Guide to VAX/VMS Software Installation* or the *Guide to VAX/VMS System Management and Daily Operations* for further information about the subjects discussed in this section.

1.3.4.1 Mandatory Post-upgrade Procedures

The following steps **must** be performed after the upgrade has finished:

- 1 Restore any SYSGEN parameters (such as SCSNODE, ALLOCLASS, SCSSYSTEMID, or VAXCLUSTER) that may have been modified to allow the upgrade to proceed.
- 2 Install the mandatory update.

After you have upgraded to VAX/VMS Version 4.4, but before you have installed any VAX/VMS options, you must immediately install the additional mandatory update.

The mandatory update is labeled as follows:

- VAX/VMS V4.4 BINRX01 Mandatory Update
- VAX/VMS V4.4 16MT9 Mandatory Update
- VAX/VMS V4.4 TU58 Mandatory Update

Use the following procedure to install the mandatory update:

- a Log in to the System Manager's account (SYSTEM).
- b Ready the mandatory update media on the device drive that you will be using.
- c Enter the following command, where *ddcu* is the name of the device on which you have mounted the update:

```
‡ @SYS$UPDATE:VMSINSTAL VMSMU044 device-name
```

The procedure prompts you for certain information (such as whether you have inserted the mandatory update and are ready to proceed). Upon completion, the procedure shuts down the system, after which you must reboot it.

Included in this update is a patch to SYS.EXE. Do not delete the existing version of SYS.EXE; use the PURGE command to remove old versions of the files after you reboot the system.

- 3 Re-install DECnet if you are running DECnet. This product must be re-installed.
- 4 Rebuild the Standalone Backup Kit. Refer to the *Guide to VAX/VMS Software Installation* for instructions.
- 5 To preserve previous layered product installations, the Version 4.4 Upgrade Procedure merges the old and new versions of the following files:

```
[SYSLIB]DCLTABLES.EXE
[SYSHLP]HELPLIB.HLB
[SYSLIB]STARLET.OLB
[SYSLIB]IMAGELIB.OLB
```

Upgrading to Version 4.4

- 6 The latest versions of the following .COM files have been placed on your new system disk. Examine these files; the original versions may have site specific changes that will be lost if you purge them. Edit the new versions as appropriate to your system.

```
[SYSMGR]SYLOGIN.COM  
[SYSMGR]RTTLOAD.COM  
[SYSMGR]SYSTARTUP.COM  
[SYSMGR]SYCONFIG.COM  
[SYSMGR]SYSHUTDOWN.COM  
[SYSMGR]STARTNET.COM  
[SYSEXE]STARTUP.COM  
[SYSEXE]SHUTDOWN.COM
```

Note for cluster upgrades: If you are upgrading a common system root (SYS\$COMMON), the new files will be in SYS\$COMMON as the latest versions and not in the specific root.

1.3.4.2 Optional Post-upgrade Procedures

This section contains optional, but recommended, procedures for you to perform on your system.

Decompressing the System Library

Many of the system libraries are shipped in a data compressed format. If you have enough disk space on your system, you should decompress the libraries to gain faster access. If you do not decompress the libraries, the performance of the HELP and LINK commands will be negatively affected.

The decompressed libraries require approximately 5000 additional blocks of disk space and the decompression process may take up to 2 hours, depending on the type of processor you are using. A general guideline for decompressing individual library files is that HELP libraries increase in size by 50 percent and OBJECT libraries by 25 percent when decompressed.

Special File Handling

After the completion of the upgrade, you may wish to remove files no longer needed. Be careful not to delete any edited files and shareable images that may be essential to the operating system.

The upgrade procedure preserves the following files:

```
[SYSEXE]NOTICE.TXT  
[SYSEXE]RIGHTSLIST.DAT  
[SYSEXE]SYSUAF.DAT
```

The following files should be deleted manually after the upgrade has completed:

```
[SYSEXE]STARTUP.UP5  
[SYSEXE]UPGRADE.KIT
```

The size of the following files may have been changed to fit the system. You may want to check these files to be sure that the sizes are appropriate:

```
[SYSEXE]SYSDUMP.DMP  
[SYSEXE]PAGEFILE.SYS  
[SYSEXE]SWAPFILE.SYS
```

Installed Images

As part of the upgrade, the file SYS\$MANAGER:VMSIMAGES.DAT is created for your site. This is a combined list of files that are installed for enhanced system performance. Some files in this list may already be installed in the SYS\$MANAGER:SYSTARTUP.COM file and should be removed from SYSTARTUP.COM.

1.4 Upgrading a VAXcluster Environment

This section describes upgrade procedures for VAXcluster environments. It is divided into the following major sections:

- Cluster-Specific Considerations
- VAX/VMS Version 4.4-Specific Considerations for Clusters
- Rolling Upgrade Procedure
- Concurrent Upgrade Procedure
- After the Cluster Upgrade

To upgrade your cluster, do the following:

- Take note of the information in Sections 1.4.1 and 1.4.2 before upgrading your cluster.
- Decide what kind of cluster upgrade you are going to do, concurrent upgrade or rolling upgrade, and perform the upgrade according to the instructions in **either** Section 1.4.3 or Section 1.4.4.
- Review the post-upgrade activities in Section 1.4.5 and complete the tasks that apply to your system.

In Section 1.4.3 (Rolling Upgrade) and Section 1.4.4 (Concurrent Upgrade), you will be instructed to perform various steps in numbered sequence. You must be sure to perform the steps in the sequence indicated. At times, you will be directed to follow procedures in other sections of this chapter. After completing a task in another section of this chapter, always return to the "cluster upgrade" section to the step where you left off and go on from there. You must continue with the instructions in the "cluster upgrade" sections until you have completed all of the steps indicated.

1.4.1 Cluster-Specific Considerations

- The high degree of sharing achieved between systems in a VAXcluster is the result of coordination at many levels of VMS. This level of coordination generally cannot be achieved across major or minor releases of VMS. Hence, all VAX/VMS members of a VAXcluster must run the same version (major and minor) of VMS. In addition, VAXcluster sites must be prepared to upgrade all VAX systems in a cluster at the same time.
- When possible, DIGITAL will endeavor to allow adjacent releases of VMS to coexist in a VAXcluster for the purpose of incrementally updating the various systems in the VAXcluster. DIGITAL recommends that this "mixed version" operation of a VAXcluster be used only to allow incremental upgrade and checkout of new versions of VMS.

Upgrading to Version 4.4

- When updating a common system root, you need to perform only one complete upgrade from one of the nodes sharing the common system root.

However, you may need to modify the console boot command files on other nodes as well as manually invoking AUTOGEN to upgrade the system configuration parameters on each node.

Alternatively, you may use the MAKEROOT command procedure to create new alternate roots for these nodes. In which case you must still modify the console boot files and manually invoke AUTOGEN.

- The term “system root” is a generic term referring to either common or private system roots.

The term “common system root” refers to a directory structure residing on a common system disk containing the system files that are shared by several processors in a cluster environment.

The term “private system root” refers to a directory structure residing in either a private, local system or a shared system disk in which the system files are not shared.

- The system being upgraded must not mount disks in use by another system, nor can any other system mount the disks in use by this system. Should either situation occur, you may irretrievably corrupt the data on the disks.

1.4.2 VAX/VMS Version 4.4-Specific Considerations

VAX/VMS Versions 4.3 and 4.4 may be intermixed in VAXcluster configurations for the purposes of incrementally applying the upgrade. Please note the following when updating incrementally:

- If you are upgrading a cluster, you must do it from a node other than the Vax 8800.
- For VAX 8800 installation, you have the option of doing either of the following: install to a private system disk and then join the cluster, or do the upgrade from a node other than the 8800 and make a new root for the 8800 to boot from. Refer to Section 1.4.5.1 of this chapter for further information.
- All systems booted from a common system root must run the same version of VAX/VMS.
- When a VAX/VMS Version 4.4 system boots in the presence of a VAX/VMS Version 4.3 system, the following informational message is printed on the system console:

```
%CSP-I-DIFSWVER, Different versions of VAX/VMS exist in cluster
```
- DIGITAL recommends the Version 4.4 upgrade be completed on all members of the cluster as quickly as possible.
- SYSGEN parameter VMDS1 must be set to 1 on all Version 4.4 nodes if a Version 4.3 system exists on the cluster. This is necessary for proper operation of the batch/print facility and RMS in a mixed version cluster. After all nodes have been upgraded to Version 4.4, the SYSGEN parameter VMDS1 should be reset to 0.

- In setting up the cluster system disk, be sure you don't have duplicate volume labels on system disks or the system will not boot when VAXcluster is greater than "1".

1.4.3 Rolling Upgrade Procedure

This section describes a method of maintaining partial system availability during an upgrade in which the old and the new VMS versions can simultaneously exist in the same VAXcluster. This procedure is applicable in VAXclusters that have multiple system roots. It is not applicable when all systems boot from a single common system root. Additional information is available in Chapter 5 of the *Guide to VAXclusters*.

You can use the rolling upgrade procedure under the following circumstances:

- When you want to upgrade and run your cluster with some nodes using Version 4.4 and other nodes using Version 4.3.
- When the Version 4.4 system disks and the Version 4.3 system disks in the cluster are separate disks.

Rolling Upgrade Procedure

A rolling upgrade is performed as follows:

- 1 Read the System Upgrade Summary section of this chapter (Section 1.3.1) to get an idea of what tasks you will be performing during the system upgrade.
- 2 Identify which processors share common system roots and which have private system roots on a private system disk. Those processors that share a common system root must be upgraded together. Those that have a private system root may be upgraded one at a time.
- 3 Depending on the configuration of your cluster, you may be asked to shut down one or more processors at a time. Check the votes and make adjustments to maintain the proper quorum that allows the cluster to continue operating properly throughout this process (the procedure is described in detail in Chapter 5 of the *Guide to VAXclusters*).
- 4 Select a system root to perform the first upgrade upon. If this system root is a common system root, shut down all but one of the processors that share that root. Do this by following the standard shutdown procedure for each node. Each time a node is shut down, invoke the command SET CLUSTER/QUORUM on one of the remaining cluster nodes. This allows one node to continue running from the common system root (assuming other nodes running from different roots supply enough votes to sustain cluster quorum). Unless proper quorum is maintained, the shutdown procedure will permanently hang the cluster. In this event, perform the following commands to free the cluster:

```

$CTRL/P
>>>H [RET]
>>>D/I 14 C [RET]
>>>C [RET]
IPC>Q [RET]
IPC>CTRL/Z
    
```

CHECKPOINT: At this point the system root selected for the upgrade should have only one processor booted from it. That processor being either the single system that normally boots from it or one of the several processors that normally boot from it.

Upgrading to Version 4.4

- 5 Perform all of the pre-upgrade procedure steps for a single system that are outlined in this chapter in the Pre-upgrade Procedure section (Section 1.3.2).
- 6 Upgrade the single system root or common system root according to the instructions found in the Performing the Upgrade section of this chapter (Section 1.3.3).

CHECKPOINT: At this point, the root has been upgraded to VAX/VMS Version 4.4 and has one processor booted from it.

- 7 Log into the upgraded processor and go on to the next step.
- 8 If a Version 4.3 system exists on the cluster, set the SYSGEN parameter VMSD1 to 1 on the upgraded (Version 4.4) system:

```
‡ RUN SYS$SYSTEM:SYSGEN
SYSGEN>USE CURRENT
SYSGEN>SET VMSD1
SYSGEN>WRITE CURRENT
SYSGEN>EXIT
```

- 9 After the upgrade is complete, you must run the AUTOGEN procedure again, as a cluster member.

Invoke AUTOGEN as follows:

```
‡ @SYS$UPDATE:AUTOGEN SAVPARAMS REBOOT
```

- 10 Wait for your system to join the VAXcluster.
- 11 Perform post-upgrade procedures according to the instructions found in the Post-upgrade Procedures section of this chapter (Section 1.3.4).
- 12 If this is a common system root, reboot the other systems that use the common system root that was just upgraded. Log into each and perform the following procedures:
 - a If a Version 4.3 system exists on the cluster, set the SYSGEN parameter VMSD1 to 1 on the Version 4.4 system:

```
‡ RUN SYS$SYSTEM:SYSGEN
SYSGEN>USE CURRENT
SYSGEN>SET VMSD1 1
SYSGEN>WRITE CURRENT
SYSGEN>EXIT
```

- b Run the AUTOGEN procedure on each. Invoke AUTOGEN as follows:

```
‡ @SYS$UPDATE:AUTOGEN SAVPARAMS REBOOT
```

- c Wait for the system to reboot and join the VAXcluster.

CHECKPOINT: At this point all systems on the common system root are running the upgraded version of VMS.

- 13 If there are additional common system roots or private system roots that have not yet been upgraded to version 4.4, you must repeat all of the tasks in this section (Steps 4 through 11) for each system root until all roots are running VAX/VMS Version 4.4.

- 14 Put the new VMB on all consoles.

Some changes have been made to an image that resides on the system console known as VMB.EXE. Normally it would be sufficient to simply copy the new image on the console. However, the size of the new image is greatly increased for VMS Version 4.4. This means that for some systems that use RX01 or TU58, there may not be enough contiguous

space on the console for the new image. A command procedure exists to create the needed space and copy the new image onto the console.

To determine if you need to update your console perform the following:

a Log into the system manager's account.

b Enter the following commands:

```
⌘ RUN SYS$SYSTEM:SYSGEN
SYSGEN>CONNECT CONSOLE
SYSGEN>EXIT
⌘ EXCHANGE DIR/SIZE CSA1:VMB.EXE
```

c If the size is at least 55 blocks, your console has the correct version of VMB.EXE. If the size is less than 55 blocks, you need to copy the correct version onto the console.

In order to update the console with the new image, invoke the console update command procedure UPDATE_CONSOLE.COM as follows:

```
⌘ @sys$update:UPDATE_CONSOLE.COM
```

If your system is an 8650, 8600, or 8200 this procedure will simply copy the new file onto your existing console.

If your system is a 11/780, 11/782, 11/785 or 11/750 this procedure will use EXCHANGE to save the contents of your existing console. It will then merge the new files on the saved copy of your console. Finally it will request that you insert a scratch medium so that it can create a new console containing the new file. Your original console will not be modified.

- 15** After all nodes have been upgraded to VAX/VMS Version 4.4, set the SYSGEN parameter VMSD1 to 0 for all nodes and reboot them.

1.4.4 Concurrent Upgrade Procedure

This section describes the upgrade procedure used when the VAXcluster cannot run more than one version of VAX/VMS simultaneously.

A concurrent upgrade is performed by bringing down the entire cluster and applying the upgraded version to each system root, one at a time. When all upgrades are complete, the cluster is brought back up running the upgraded version.

The concurrent upgrade procedure should be used under the following circumstances:

- When all nodes in a cluster must run the same version of VAX/VMS simultaneously. In the specific case of upgrading to Version 4.4 of VAX/VMS, a rolling upgrade is permitted from Version 4.3 to Version 4.4.
- When the cluster members have a common system disk.

Upgrading to Version 4.4

Concurrent Upgrade Procedure

A concurrent upgrade is performed as follows:

- 1 Make note of the votes and quorum. Upon completion of the upgrade, they must be reset to their original values (the procedure is described in full detail in Chapter 5 of the *Guide to VAXclusters*).

Note the current values for votes and quorum on each processor as follows:

```
⚡ RUN SYS$SYSTEM:SYSGEN
SYSGEN>USE CURRENT
SYSGEN>SHOW VOTES
SYSGEN>SHOW QUORUM
SYSGEN>EXIT
```

- 2 Identify which processors share common system roots and which have private system roots on a private system disk. Those that share a common system root must be upgraded together. Those that have a private system root may be upgraded one at a time.
- 3 Shut down the entire cluster using the standard shutdown procedure.
- 4 Select a system root to perform the first upgrade upon.
- 5 Perform a conversational boot (see *VAX/VMS System Manager's Reference Manual* on a single VAX system that boots from the selected root and set the votes and quorum values to 1 as follows:

```
SYSBOOT>SET VOTES 1
SYSBOOT>SET QUORUM 1
SYSBOOT>CONTINUE
```

- 6 Perform the Version 4.4 upgrade procedure as described in the Upgrading a System section (Section 1.3) of this chapter. You will apply the upgrade to the root from which the system is booted.
- 7 After the upgrade procedure has completed, you must run the AUTOGEN procedure for a second time.

Invoke AUTOGEN as follows:

```
⚡ @SYS$UPDATE:AUTOGEN SAVPARAMS SHUTDOWN
```

- 8 Select another system root and repeat Steps 4 through 7 until the upgrade has been applied to each system root in the cluster.

Note: In the case of common system roots where several processors boot from the same disk, this procedure need only be performed one time for the common root.

- 9 Bring up the entire cluster according to your normal operating procedures. The entire cluster will now be running the upgraded version of VMS.
- 10 You must now run AUTOGEN on all systems that boot from a common system root and that have not directly performed an upgrade.

Invoke AUTOGEN as follows:

```
⚡ @SYS$UPDATE:AUTOGEN SAVPARAMS REBOOT
```


11 Put the new VMB on all consoles.

Some changes have been made to an image that resides on the system console known as VMB.EXE. Normally it would be sufficient to simply copy the new image on the console. However, the size of the new image is greatly increased for VMS Version 4.4. This means that for some systems that use RX01 or TU58, there may not be enough contiguous space on the console for the new image. A command procedure exists to create the needed space and copy the new image onto the console.

To determine if you need to update your console perform the following:

- a Log into the system manager's account.
- b Enter the following commands:

```
‡ RUN SYS$SYSTEM:SYSGEN
SYSGEN>CONNECT CONSOLE
SYSGEN>EXIT
‡ EXCHANGE DIR/SIZE CSA1:VMB.EXE
```

- c If the size is at least 55 blocks, your console has the correct version of VMB.EXE. If the size is less than 55 block you need to copy the correct version onto the console.

In order to update the console with the new image, invoke the console update command procedure UPDATE_CONSOLE.COM as follows:

```
‡ @sys$update:UPDATE_CONSOLE.COM
```

If your system is an 8650, 8600, or 8200 this procedure will simply copy the new file onto your existing console.

If your system is a 11/780, 11/782, 11/785 or 11/750 this procedure will use EXCHANGE to save the contents of your existing console. It will then merge the new files on the saved copy of your console. Finally it will request that you insert a scratch medium so that it can create a new console containing the new file. Your original console will not be modified.

1.4.5 After the Cluster Upgrade

This section contains procedures for you to perform after you have finished upgrading your cluster.

1.4.5.1 Creating Alternate Roots on a Common System Disk

If you chose to create a common system root during the upgrade procedure, you may want to perform the procedures in this section so that other processors can also boot from the common system root.

In a VAXcluster that features a common system disk, the MAKEROOT.COM command procedure is used to create system roots for nodes other than the first node of the cluster. To invoke MAKEROOT, enter the following command:

```
‡ @SYS$MANAGER:MAKEROOT
```

Note, MAKEROOT will abort if the system disk is not a cluster-common system disk, or if the SYSGEN parameter VAXCLUSTER is 0.

Upgrading to Version 4.4

When MAKEROOT executes, it requests the name of the new system root. Enter the root name using the form SYSx, where x is a hexadecimal digit between 1 and D (for example, SYS1 or SYSA). Note, system roots SYSE and SYSF are reserved for other system functions.

You may not specify the root of the currently running system (usually SYS0). If any other existing system root is specified, the system queries if you want to modify the existing system root. If YES, MAKEROOT deletes the following root files:

- [SYSEXE]MODPARAMS.DAT
- [SYSEXE]VAXVMSSYS.PAR
- [SYSMGR]VMSIMAGES.DAT

If a SYSCOMMON directory exists in the directory tree, it is also deleted. Note, MAKEROOT does not check the format of the existing system root. DIGITAL recommends you delete the directory tree or choose a different root.

Next, the MAKEROOT command queries for the new node name and its SCSSYSTEMID. The node name can be no more than six characters. Once the node name and the value of SCSSYSTEMID are provided, MAKEROOT queries for the size of the page file and swap file of the new root. The values you provide are subsequently used by AUTOGEN.

Finally, MAKEROOT creates the new directory tree, the page file, and the swap file. It also generates a VAXVMSSYS.PAR file for the new root using the SYSGEN parameters of the currently running node as the basis for the new node. When completed, the system displays a series of messages that describe how to build the console media for the new system. Once you have built the new console media, the system must be rebooted. When the system is running, invoke AUTOGEN as follows:

```
Ⓢ @SYS$UPDATE:AUTOGEN SAVPARAMS REBOOT
```

1.4.5.2 Upgrading DECnet

With VMS Version 4.4, it is possible to share components of the DECnet permanent database with nodes in a homogeneous VAXcluster. These components are created in SYS\$SYSTEM:, which is SYS\$SPECIFIC:[SYSEXE] in a normally configured node. For example, the permanent object database should be SYS\$SPECIFIC:[SYSEXE]NETOBJECT.DAT. If the permanent object database is identical on every cluster node, it can be shared as follows:

- 1 Rename (or move) the permanent object database on one node to the common system root. For example:

```
Ⓢ RENAME SYS$SPECIFIC:[SYSEXE]NETOBJECT.DAT -  
_Ⓢ SYS$COMMON:[SYSEXE]NETOBJECT.DAT  
or  
Ⓢ COPY SYS$SPECIFIC:[SYSEXE]NETOBJECT.DAT -  
_Ⓢ SYS$COMMON:[SYSEXE]NETOBJECT.DAT
```

- 2 Delete (or rename) the permanent object database from the private system root on each node in the VAXcluster. For example,

```
Ⓢ DELETE SYS$SPECIFIC:[SYSEXE]NETOBJECT.DAT;*  
or  
Ⓢ RENAME SYS$SPECIFIC:[SYSEXE]NETOBJECT.DAT;* -  
SYS$SPECIFIC:[SYSEXE]NETOBJECT.OLD;*
```

Upgrading to Version 4.4

Prior to VMS Version 4.4, the permanent node database was comprised of a single file, SYS\$SYSTEM:NETNODE.DAT. Because this file contains executor information, it is not directly amenable to sharing in a VAXcluster. In VMS Version 4.4, the permanent node database is comprised of two files, SYS\$SYSTEM:NETNODE_LOCAL.DAT that contains records defining the executor and loop nodes, and SYS\$SYSTEM:NETNODE_REMOTE.DAT that contains records defining remote nodes. NETNODE_LOCAL.DAT should never be moved to the common system root because the information it contains is specific to one node only; NETNODE_REMOTE.DAT, however, is designed to reside in the common system root.

Conversion of NETNODE.DAT to NETNODE_LOCAL.DAT and NETNODE_REMOTE.DAT is accomplished by executing NCP and referencing any component of the permanent database; if one or both of the components are missing, NETNODE.DAT will be read to build the missing components.

The conversion may take seconds or several minutes, depending on the number of nodes to be converted. This conversion is accomplished automatically as part of Phase 5 of the VMS upgrade procedure. Note, however, the conversion will have taken place only for the single node in the homogeneous VAXcluster that actually performed the upgrade. For the other nodes in the VAXcluster, the conversion will take place automatically when NETCONFIG.COM is used or when DECnet is started.

If the permanent remote node database is to be shared on the VAXcluster, the following procedure can be applied after one node has been upgraded and the remaining nodes have been rebooted:

- 1 On the node that performed the upgrade, rename the permanent remote node database to the common system root as follows:

```
⌘ RENAME SYS$SPECIFIC:[SYSEXE]NETNODE_REMOTE.DAT -  
_⌘ SYS$COMMON:[SYSEXE]NETNODE_REMOTE.DAT
```

- 2 On each other node in the homogeneous VAXcluster, create the permanent local node database as follows:

```
⌘ RUN SYS$SYSTEM:NCP  
NCP> LIST EXECUTOR
```

- 3 Once each node has a permanent local node database, the old permanent node database can be removed as follows:

```
⌘ DELETE SYS$SPECIFIC:[SYSEXE]NETNODE.DAT;*
```

For additional information on the components comprising the permanent database, see Chapter 5 of the *VAX/VMS Networking Manual*.

2 New and Changed Features

This chapter contains information pertaining to the new features that have been added for VAX/VMS Version 4.4. A brief description of each new feature is provided, including a reference to where more information can be found on the topic.

Topics in this chapter can be found under the following categories:

- Section 2.1—General User Information
- Section 2.2—System Manager Information
- Section 2.3—Application Programmer Information
- Section 2.4—System Programmer Information

To find specific topics, consult the index in the back of this manual.

2.1 General User Information

This section contains information about new features in Version 4.4 of interest to the general user.

2.1.1 Command Line Recall Capability—Expanded

For interactive utilities and layered products that use the VAX/VMS screen management software, you can recall up to the last twenty command lines by entering `CTRL/B` or by using the up-arrow and down-arrow keys. Examples of VAX/VMS utilities that permit this feature are Mail, Debug, Show Cluster, SDA, EDT Editor and VAXTPU Editor. VAX C Run-Time Library support also permits this feature.

2.1.2 DCL (Digital Command Language) Commands—New Commands

New commands are:

- CALL
- GOSUB
- RETURN
- SET SYMBOL/SCOPE
- SUBROUTINE
- ENDSUBROUTINE

See the *VAX/VMS DCL Dictionary* for more information about these commands.

New and Changed Features

2.1.3 File Specifications and Logical Names—Hyphen Permitted

The hyphen (-) is permitted in the file name, file type, and directory fields of a DCL file specification. Hyphens may also be used in a logical name that appears as an unpunctuated file specification. Hyphens cannot be used in node names or device names.

Since the hyphen is also the DCL command-continuation character, entering a file name ending with a hyphen can be awkward. Therefore, creating files with a trailing hyphen in the name or type field is not recommended.

2.1.4 VAX Text Processing Utility (VAXTPU)—Changes

This section outlines changes which have been made to the VAX Text Processing Utility (VAXTPU).

2.1.4.1 Recompiling Section Files

In making VAXTPU faster and more functional, the format of the section files were changed. All section files must be rebuilt.

2.1.4.2 Default Section File Type Change

The default file type for VAXTPU section files was changed with this release. The default file type is now TPU\$SECTION; in previous versions, it was GBL.

2.1.4.3 EVE\$INIT_KEY and EVE\$CLEAR_KEY in EVE

The internal, pre-defined EVE procedures *eve\$init_key* and *eve\$clear_key* are no longer used with VAX/VMS Version 4.4. If you used these procedures to extend the EVE interface, substitute the VAXTPU built-in procedures, DEFINE_KEY and UNDEFINE_KEY, respectively.

Be aware of key maps and key-map lists when defining keys. EVE does not use VAXTPU's normal default key map, TPU\$KEY_MAP. The key map you probably want to use is EVE\$USER_KEYS. EVE\$USER_KEYS is the default key map in EVE which has precedence over EVE's other key maps. Other key maps in EVE include either EVE\$VT100_KEYS or EVE\$VT200_KEYS for the keypad keys, and EVE\$STANDARD_KEYS for all other keys.

2.1.4.4 Callable Interface

The callable interface has the following two changes.

When calling the user I/O routine with the code TPU\$K_GET, VAXTPU now passes a valid dynamic string descriptor as the DATA parameter. Your I/O routines should pass records to VAXTPU with the VAX/VMS Run Time Library string copy routines. Otherwise, you may get an access violation error (ACCVIO) when VAXTPU attempts to free the memory allocated to the string.

The global symbols TPU\$COMMAND_TABLE, TPU\$FACILITY_NAME, and TPU\$MESSAGE_FLAGS in the callable interface are no longer used.

2.1.4.5 GET_INFO (SYSTEM, "timed_message")

In previous versions, when the timer was set to OFF with the built-in procedure SET (TIMER, OFF), the built-in procedure GET_INFO (SYSTEM, "timed_message") returned a null string. With this release, the string that was last specified with the built-in procedure SET (TIMER,...) is returned. If no string was specified, the default string "working" is returned.

2.2 System Manager Information

This section contains information about the VAX/VMS Version 4.4 Release intended for system managers.

2.2.1 Networking—New and Changed Features

New networking features for VMS Version 4.4 include the following:

- DECnet-VAX supports the ability to identify a VAXcluster as a single node by means of an alias, while retaining the ability to address each node in the cluster individually. Some or all nodes in the cluster can assume an alias node identifier (name or address). New NCP parameters supporting the alias are the executor parameters ALIAS NODE, ALIAS INCOMING, and ALIAS MAXIMUM LINKS, and the object parameters ALIAS INCOMING and ALIAS OUTGOING.
- Files composing the permanent database can now be shared between nodes in a VAXcluster. The Version 4.3 permanent node file NETNODE.DAT will be converted during the VMS Version 4.4 upgrade to two files: NETNODE_LOCAL.DAT and NETNODE_REMOTE.DAT.
- A new Ethernet circuit device, the DELUA, is similar in function to the DEUNA.
- A new NCP parameter, TRANSMIT PIPELINE, for DMR11 lines, provides for more efficient use of satellite links.
- Support has been added for multiple networks in VAX PSI configurations. A VAX PSI system can now be configured for more than one DTE and for several network connections to several PSDNs. VAX PSI also supports DCE to DTE operation, enabling the X.25 protocol module to be used for back-to-back operations.

For complete lists of the new networking features, see the “New and Changed Features” sections of the *VAX/VMS Networking Control Program Reference Manual* and the *VAX/VMS Networking Manual*. Note that the *VAX/VMS Networking Manual* was previously entitled the *Guide to Networking on VAX/VMS*.

2.2.2 Cluster Node Names—New Feature

A cluster manager is now able to treat the nodes in a homogeneous VAXcluster as a single node in a DECnet network by specifying a cluster node name. A whole VAXcluster or some of the nodes in a cluster can be represented by a single alias node identifier. This mechanism allows users on DECnet nodes outside the cluster to access cluster resources without knowing which individual cluster nodes exist or which one are active. Use of the cluster alias node name and address never precludes the use of the individual node name and address. Thus, a remote node can address the cluster as a single node and address any cluster member individually.

This information will be incorporated into the *Guide to VAXclusters* in a future revision. For more information about cluster alias node identifiers, see the *VAX/VMS Networking Manual*.

New and Changed Features

2.2.3 Show Cluster Utility—New Commands and Changed Features

The commands and features affected are outlined in the following sections.

New Features for the Show Cluster Utility

The Show Cluster Utility for Version 4.4 contains the following new features:

- New commands:
 - DEFINE/KEY
 - DESELECT
 - MOVE
 - PAN
 - REFRESH
 - SAVE
 - SCROLL
 - SELECT
 - SET AUTO_POSITIONING
 - SET FUNCTION
 - WRITE
- Definable keys and a default keypad.
- Three display windows, each containing related classes.
- Each window can be manipulated independently.
- Command procedures can be nested up 16 levels deep.
- One display report instead of two. (Consequently, the SHOW command is now obsolete.)

For more information, see the *VAX/VMS Show Cluster Utility Reference Manual*.

/REPORT Qualifier Unsupported

The /REPORT qualifier is now unsupported. The information that was previously separated into two distinct reports, CLUSTER and LOCAL_PORTS, can now be displayed on the same screen by using the ADD command.

The default display of the SHOW CLUSTER command remains what was formerly the CLUSTER report. If it is desirable to reproduce the LOCAL_PORTS report, as was possible using the /REPORT=LOCAL_PORTS qualifier, simply create an initialization file containing the following commands:

```
REMOVE SYSTEMS, MEMBERS
ADD LOCAL_PORTS, ERRORS
```


2.2.4 System Security—New Command and Attribute Features

New features include a new DCL command, SET RIGHTS_LIST and a new attribute, DYNAMIC.

SET RIGHTS_LIST adds and removes identifiers from the process and system rights list. You can assign the DYNAMIC attribute to identifiers to enable unprivileged users to add or remove identifiers they hold from their process rights list.

For more information on changes to the security system services, see the “New and Changed Features” section of the *VAX/VMS System Services Reference Manual*.

2.2.5 BATCH/PRINT Facility—New Features

The following new and changed features are documented in Section 9 of the *VAX/VMS System Manager's Reference Manual* for Version 4.4. Note that this manual was previously entitled *Guide to VAX/VMS System Management and Daily Operations*.

- The DCL command START/QUEUE/MANAGER has a new /[NO]RESTART qualifier. This qualifier causes the queue manager to automatically restart on recovery from a job controller abort. In addition, batch and output queues are restored to the status that existed prior to the interruption of service. The default is /NORESTART. For more information, see the *VAX/VMS DCL Dictionary*.
- You can now define a queue-specific default form for a printer, terminal, or server queue. To do this, first define a form using the DEFINE/FORM command. Then use the new FORM=type keyword with any of the following commands:

```
INITIALIZE/QUEUE/DEFAULT=(FORM=type)
START/QUEUE/DEFAULT=(FORM=type)
SET QUEUE/DEFAULT=(FORM=type)
```

A job submitted without an explicit form name in the PRINT command line will be processed using the default form specified for the queue.

- The /FORM qualifier has been renamed to /FORM_MOUNTED for the following commands:

```
INITIALIZE/QUEUE/FORM_MOUNTED=type
START/QUEUE/FORM_MOUNTED=type
SET QUEUE/FORM_MOUNTED=type
```

The FORM_MOUNTED qualifier associates the paper stock of a form with that of the output queue. The stock types must match or the job will enter a pending state.

- The new /NOAFTER qualifier can be used with the SET QUEUE/ENTRY command to immediately release a job held by previously specifying the /AFTER qualifier.
- The new /[NO]PAGE_SETUP qualifier of the SET QUEUE/ENTRY command specifies one or more device control library modules, which perform special printer functions before the printing of each page.

New and Changed Features

2.2.6 VAX 8600 Memory—Additional Error Logging

The VAX/VMS operating system now supports additional error logging information for the two kinds of VAX 8600 memory. The upper byte of the MSTAT2 longword on a machine check stack frame has been reserved for software. VAX/VMS uses this byte to include a 4-bit nibble. The low three bits give the memory type where an error occurred if the machine check was due to a memory error. The upper bit is "on" if the low three bits contain valid information. In all other cases, it is "off".

2.2.7 Authorize Utility—Changes

This section contains information pertaining to the Authorize Utility.

2.2.7.1 /ATTRIBUTES—New Keyword

The Authorize Utility has a new keyword for the /ATTRIBUTES qualifier. You can specify the [NO]DYNAMIC keyword with the following commands:

```
ADD/IDENTIFIER/ATTRIBUTES
GRANT/IDENTIFIER/ATTRIBUTES
MODIFY/IDENTIFIER/ATTRIBUTES
```

Specifying the [NO]DYNAMIC keyword indicates whether unprivileged holders of the identifiers may add or remove them from the process rights list. The default is NODYNAMIC.

This information will be added to the *VAX/VMS Authorize Utility Reference Manual* in a future release.

2.2.7.2 /ACCESS Qualifier—Enhanced

The syntax string for the /ACCESS qualifier to the MODIFY command has been enhanced to allow more readable, flexible usage. The following commands produce identical results.

```
UAF> MODIFY SAM /ACCESS=(primary, 2-3, 5, secondary, 8-12)
UAF> MODIFY SAM /ACCESS="Primary: 2-3, 5; Secondary: 8-12"
UAF> MODIFY SAM /ACCESS=(p,2,s,8,p,3,s,9,p,5,s,10-12)
UAF> MODIFY SAM /ACCESS="2-3 SEC 8-12 PRIM 5"
```

2.2.7.3 /DEFPRIVILEGES and /PRIVILEGES Qualifiers

You can specify the keyword [NO]ALL for the /DEFPRIVILEGES and /PRIVILEGES qualifiers to disable/enable all user privileges.

2.2.7.4 Secondary Passwords—Change

Beginning with Version 4.2, users cannot initially give themselves secondary passwords. The initial setting of the secondary password must be done by the system manager using the Authorize Utility. The reason for this change is to protect careless users who leave their terminal sessions unattended.

In earlier versions of VAX/VMS, anyone could essentially render an account useless by simply adding a secondary password that the account's owner did not know. If a user now tries to initiate a secondary password, the system will respond as follows:

```
‡ SET PASSWORD/SECONDARY
%SET-F-PWD2NOTSET, system manager must initially set secondary passwords
```

2.2.7.5 AUTOLOGIN Flag—New Feature

A flag named AUTOLOGIN is added to the flags field in the user authorization file (SYSUAF). The flag is set by specifying the qualifier /FLAGS=AUTOLOGIN to one of the following Authorize Utility commands: ADD, MODIFY, or COPY. When set, it makes the account available only by using the autologin mechanism. The following forms of access are disabled:

- Login by any terminal, LAT connection, or SET HOST involving presentation of username and password
- Access by DECnet task using explicit access control

The following forms of access remain permitted:

- Interactive login by the autologin mechanism
- Batch jobs
- Proxy access by DECnet task

2.2.8 VAX Text Processing Utility (VAXTPU)—Changes

The following changes to the VAX Text Processing Utility (VAXTPU) affect system managers.

2.2.8.1 VAXTPU Packaging

Previously, VAXTPU was packaged in one shareable image, TPUSHR.EXE. With this release, the screen management routines are placed in their own shareable image, TPU\$CCTSHR.EXE. Install each image as follows:

```
INSTALL> SYS$SHARE:TPUSHR.EXE /OPEN/HEADER/SHARE
INSTALL> SYS$SHARE:TPU$CCTSHR.EXE /OPEN/HEADER/SHARE
```

2.2.8.2 Changing the Editing Interface from EVE to the EDT Keypad Emulator

To change the default editing interface from EVE to the EDT Keypad Emulator, copy the section file SYS\$LIBRARY:EDTSECINI.TPU\$SECTION to SYS\$LIBRARY:TPUSECINI.TPU\$SECTION.

2.2.8.3 Using Section Files

Section files are now installable as shared images. At the INSTALL utility, enter the following:

```
INSTALL> SYS$SHARE:TPUSECINI.TPU$SECTION /OPEN/HEADER/SHARE
```

The preferred method of invoking a section file other than the default section file is to define the logical TPUSECINI to point to the appropriate section file. For example:

```
* DEFINE TPUSECINI ddn:[dir]mysection.TPU$SECTION
```

where

dd is the device code and *n* is its unit number

dir is the directory name

mysection is the file name for the section file to be invoked.

New and Changed Features

2.2.9 RMS—Now Supports Full Shared Sequential Files

RMS now supports shared access to any form of sequential organization file. However, this added support has a potential effect on mixed-version cluster operations. Specifically, if two cluster nodes, one running Version 4.4 software the other Version 4.3, attempt shared access of a sequential file, the Version 4.3 node may receive an RMS\$_FLK error code if the Version 4.4 node opens the file first.

In order to reduce the impact of this change, a special SYSGEN parameter, VMSD1, is used to notify the 4.4 node that a mixed version cluster is possible. By setting this parameter to 1, potential RMS\$_FLK errors are avoided. Note, even with this parameter set to 1 (mixed cluster mode), RMS still supports sequential file sharing. Since the mixed cluster mode setting reduces RMS performance of shared sequential files, you should set the VMSD1 SYSGEN parameter to 0 when VAX/VMS Version 4.3 nodes are no longer part of the cluster.

2.2.10 Clusters—Limiting Access to Layered Products

It is now possible to limit access to a layered product to one or more nodes in a cluster if the layered product is placed in a common system directory. The method utilizes the new access control list (ACL) features and a system rights identifier that is created during the startup phase of the boot process.

The identifier created at boot time is **SYSSNODE_nodename** where **nodename** is the name defined by the SYSGEN parameter SCSNODE. Because each node in the cluster has a unique SCSNODE name, each node has a unique identifier.

Note: An identifier is not created for a particular node in the cluster if any of the following are true:

- 1 The SYSGEN parameter SCSNODE is not defined.
- 2 The rights list file RIGHTSLIST.DAT is not created in your site-specific command procedure (SYSS\$MANAGER:SYSTARTUP.COM). See the *Authorize Utility Reference Manual* for information about how to create this file.
- 3 The command LOGOUT is executed from SYSTARTUP.COM. If this is the current behavior, the command must be changed to EXIT to return control to STARTUP.COM.

When a user logs in, the node-specific identifier is associated with the process. Invoking SHOW PROCESS/PRIVILEGE on NODE1 might yield the following message:

```
Process privileges:
SYSPRV may access objects via system protection
CMKRNL may change mode to kernel
OPER operator privilege

Process rights identifiers:
SYSSNODE_NODE1
```

The ACL commands used must limit access to the key images of the chosen layered product. For example, if the product were a language, the image that compiles the program would be the file to which access should be restricted. The SET FILE/ACL=(IDENTIFIER) command is used to create the access

control list for the file specified; a SHOW ACL command displays the access control list.

It should be mentioned that not all files on a layered product need to be restricted.

The following example reflects the necessary commands for limiting node access to a layered product.

```

$! Allow access to the FORTRAN compiler on NODE1 in the cluster.
$ SET FILE/ACL=(IDENTIFIER=*, ACCESS=NONE) SYS$SYSTEM:FORTRAN.EXE
$ SET FILE/ACL=(IDENTIFIER=SYS$NODE_NODE1, ACCESS=EXECUTE) SYS$SYSTEM:FORTRAN.EXE
$! NOTE: The first SET FILE/ACL command specifies that no one has access
$!       to the FORTRAN image. The second command specifies that only processes
$!       with the identifier SYS$NODE_NODE1 can access the image. Thus,
$!       only those users who can log into NODE1 may use FORTRAN.
$! Limit access to the other key FORTRAN images.
$ SET FILE/ACL=(IDENTIFIER=*, ACCESS=NONE) SYS$MESSAGE:FORTERR1.EXE
$ SET FILE/ACL=(IDENTIFIER=SYS$NODE_NODE1, ACCESS=EXECUTE) SYS$MESSAGE:FORTERR1.EXE
$ SET FILE/ACL=(IDENTIFIER=*, ACCESS=NONE) SYS$MESSAGE:FORTERR2.EXE
$ SET FILE/ACL=(IDENTIFIER=SYS$NODE_NODE1, ACCESS=EXECUTE) SYS$MESSAGE:FORTERR2.EXE
$! Show the access control list for the FORTRAN compiler.
$ SHOW ACL SYS$SYSTEM:FORTRAN.EXE
Object type: file, Object name: DISK$: [SYSEXE]FORTRAN.EXE;1,
on 01-01-1985 10:00:00.00
      (IDENTIFIER=SYS$NODE_NODE1,ACCESS=EXECUTE)
      (IDENTIFIER=*,ACCESS=NONE)

$! Allow access to the layered product BASIC for two nodes in the cluster.
$ SET FILE/ACL=(IDENTIFIER=*) SYS$SYSTEM:BASIC.EXE
$ SET FILE/ACL=(IDENTIFIER=SYS$NODE_NODE1+SYS$NODE_NODE2, ACCESS=EXECUTE) SYS$SYSTEM:BASIC.EXE
$ SET FILE/ACL=(IDENTIFIER=*) SYS$MESSAGE:BASICMSG.EXE
$ SET FILE/ACL=(IDENTIFIER=SYS$NODE_NODE1+SYS$NODE_NODE2, ACCESS=EXECUTE) SYS$MESSAGE:BASICMSG.EXE

```

The following list contains various layered products and their associated key image file names.

Table 2-1 Layered Products with File Names

Layered Product	Key Image File Name
VAX ADA	SYS\$SYSTEM:ADA.EXE SYS\$SYSTEM:ADA\$FROM_CDD.EXE SYS\$SYSTEM:ADA\$STAT.EXE
VAX APL	SYS\$SYSTEM:APL.EXE SYS\$LIBRARY:APLTAP.EXE SYS\$MESSAGE:APLMSG.EXE
VAX BASIC	SYS\$SYSTEM:BASIC.EXE SYS\$SYSTEM:BASICMSG.EXE
VAX BLISS-16	SYS\$SYSTEM:BLISS16.EXE
VAX BLISS-32	SYS\$SYSTEM:BLISS32.EXE
VAX C	SYS\$SYSTEM:VAXC.EXE SYS\$MESSAGE:VAXCCRERR.EXE SYS\$MESSAGE:VAXCERR.EXE SYS\$MESSAGE:VAXCVCERR.EXE

New and Changed Features

Table 2–1 (Cont.) Layered Products with File Names

Layered Product	Key Image File Name
VAX CDD	SYSS\$SYSTEM:CDDL.EXE SYSS\$SYSTEM:CDDV.EXE SYSS\$SYSTEM:DMU.EXE SYSS\$MESSAGE:CDDVEXC.EXE SYSS\$MESSAGE:CDDVEXC.EXE SYSS\$MESSAGE:CDDLEXC.EXE SYSS\$MESSAGE:DMUEXC.EXE
VAX COBOL	SYSS\$SYSTEM:COBOL.EXE SYSS\$SYSTEM:COB11T.EXE
VAX DIBOL	SYSS\$SYSTEM:DIBOL83.EXE SYSS\$MESSAGE:DIBOL83MSG.EXE SYSS\$MESSAGE:DBLRTLMSG.EXE SYSS\$SYSTEM:DBLSORT.EXE SYSS\$SYSTEM:DBLSORT2.EXE SYSS\$SYSTEM:DBLSTATUS.EXE SYSS\$SYSTEM:DBLMSGMGR.EXE SYSS\$SYSTEM:DBLISMUTL.EXE SYSS\$LIBRARY:DBLRTL.EXE
VAX DSM	SYSS\$SYSTEM:DSM.EXE SYSS\$MESSAGE:DSMJRN.EXE SYSS\$MESSAGE:DSMMJC.EXE
VAX FORTRAN	SYSS\$SYSTEM:FORTTRAN.EXE SYSS\$MESSAGE:FORTERR1.EXE SYSS\$MESSAGE:FORTERR2.EXE
VAX LISP	SYSS\$COMMON:[VAXLIS]LISP.EXE
VAX LSE	SYSS\$SYSTEM:LSEEDIT.EXE SYSS\$MESSAGE:LSEMSG.EXE SYSS\$LIBRARY:LSESHR.EXE
VAX PASCAL	SYSS\$SYSTEM:PASCAL.EXE SYSS\$MESSAGE:PASCALER1.EXE SYSS\$MESSAGE:PASCALER2.EXE
VAX PL/I	SYSS\$SYSTEM:PLIG.EXE SYSS\$MESSAGE:PLIGERR1.EXE SYSS\$MESSAGE:PLIGERR2.EXE SYSS\$MESSAGE:PLIGERR3.EXE
VAX RPG II	SYSS\$SYSTEM:RPG.EXE

2.2.11 VAX–11/730 Dual-RL02 System—No Longer Supported

The VAX–11/730 Dual-RL02 system is no longer supported as of Version 4.2. Documentation on the Dual-RL02 system will remain in the VAX/VMS documentation set until the next major release.

2.2.12 System Generation Utility (SYSGEN)—New Qualifier

CONNECT CONSOLE Command: /Remote Qualifier

The /REMOTE qualifier has been added to CONNECT CONSOLE. This qualifier enables a remote diagnostic port for a second console or terminal connected to a VAX 8600.

2.2.13 MOUNT/CACHE=TAPE_DATA

The MOUNT command /CACHE=option qualifier has a new option, TAPE_DATA. The /CACHE=TAPE_DATA qualifier enables the write cache for a tape device if the tape controller supports a write cache. /NOCACHE is the default for mounting on tape devices. If the tape controller does not support a write cache, the option is ignored.

Note that the other options for the /CACHE=option qualifier pertain only to disks, while the TAPE_DATA option is used only with magnetic tapes.

```

$ MOUNT/CACHE=TAPE_DATA MUA0: TAPE
%MOUNT-I-MOUNTED, TAPE mounted on _NODE$MUA0:

```

This command mounts the volume TAPE on device MUA0 and instructs MOUNT to enable the tape controller's write cache for MUA0:.

2.2.14 Monitor Utility—New Commands and Changed Features

Version 4.4 MONITOR includes the following new functions:

- A new CLUSTER class¹ that permits live display and recording of significant clusterwide performance data for up to 16 member systems. Two screen formats are available.
- A new CONVERT command that converts to Version 4.4 format recording files produced with previous MONITOR versions.
- A new /NODE command qualifier¹ that allows users to request performance data for any or all classes for up to 16 VAXcluster member systems.
- The I/O Request Queue Length item of the DISK class is now sampled every second regardless of the collection interval value. This change yields queue length statistics with a consistently low sampling error.

¹ Requires that DECnet-VAX be installed.

New and Changed Features

2.2.15 AUTOGEN Command Procedure—New Features

This section describes Version 4.4 changes to the AUTOGEN.COM command procedure.

Calculation of LRPSIZE Parameter

In Version 4.4, AUTOGEN sets a value of 1504 for the LRPSIZE parameter if Ethernet is present on the system. This value is set on the assumption that Ethernet is the preferred DECnet communication mode. If you are using another mode (primarily CI, for example), edit the file SYS\$SYSTEM:MODPARAMS.DAT and specify a value of 576 for LRPSIZE. When you rerun AUTOGEN, all related values will be set appropriately.

Calculation of VAXCLUSTER Parameter

AUTOGEN now calculates the correct value for the VAXCLUSTER parameter. It is no longer necessary to set the value explicitly in MODPARAMS.DAT.

Calculation of Primary Page and Swap File Sizes

If, to meet special needs on your system, you have manually allocated secondary page or swap file space, you should specify a value of 0 for the PAGEFILE and SWAPFILE parameters in MODPARAMS.DAT. AUTOGEN will then not attempt to recalculate file sizes.

2.2.16 Show Cluster—Arrow Keys Function Changed

The default function of the arrow keys when Show Cluster is run in the /CONTINUOUS mode has been changed in VAXVMS Version 4.4 to provide command line recall and command line editing. To change the function of the arrow keys to pan the display, or to scroll or move windows within the display, the SET FUNCTION command must be used.

An initialization file can be created that will set the function of the arrow keys such that their behavior mimics the behavior in versions of Show Cluster prior to VAX/VMS Version 4.4. To do this, the initialization file should contain one of the following sequences of commands:

```
SET FUNCTION PAN
      or
SET FUNCTION SCROLL
SELECT
```

2.3 Application Programmer Information

This section contains information about new and changed features available to application programmers.

2.3.1 Linker Utility—Debugging Permitted for Shareable Images

You can now link a shareable image by issuing the following command line:

```
‡ LINK/SHAREABLE/DEBUG image-name,...
```

The qualifiers `/[NO]TRACEBACK` and `/DEBUG` will be processed for a shareable image exactly as they are for an executable image. Previously, the `/DEBUG` qualifier was prohibited and the `/[NO]TRACEBACK` qualifier was ignored when linking a shareable image.

2.3.2 Debugger—New Features

New debugger features, summarized below, are described in the *VAX/VMS Debugger Reference Manual*.

2.3.2.1 Screen Mode Enhancements

Screen mode enhancements are as follows:

- New PROMPT predefined display.
- New display attributes for SELECT command (`/INPUT`, `/ERROR`, `/PROGRAM`, `/PROMPT`).
- SET WINDOW, SET DISPLAY, and DISPLAY commands let you divide windows vertically.
- New built-in window definitions.
- New built-in symbols `%PAGE` and `%WIDTH`.
- New MOVE, EXPAND, and EXTRACT commands.
- Displays are now dynamic by default (display windows are resized as you change screen height or width).
- New qualifiers for DISPLAY and SET DISPLAY commands: `/[NO]DYNAMIC`, `/[NO]POP`, `/[NO]PUSH`.
- SHOW DISPLAY and SHOW WINDOW commands accept list of parameters, wildcards, and `/ALL` qualifier.
- New keypad key definitions.

2.3.2.2 Other New Features and Changes

Other new features and commands are as follows:

- The debugger supports VAX DIBOL and VAX SCAN.
- You can debug shareable images (new (SET, SHOW, CANCEL) IMAGE commands).
- New ATSIGN commands (SET, SHOW) let you set/display the default file specification for command procedures.
- New EDITOR commands (SET, SHOW) let you define/display the editor invoked by the EDIT command.
- New SHOW STACK command provides detailed information about the call stack.
- You can qualify the SET BREAK, SET TRACE, and STEP commands with the `/[NO]SHARE` and `/[NO]JSB` qualifiers.

New and Changed Features

- You can control the interpretation of data in untyped locations by specifying EXAMINE/TYPE=exp, DEPOSIT/TYPE=exp, and SET TYPE TYPE=exp.
- New default scope for symbol lookup (equivalent to SET SCOPE 0,1,2,3, . . . ,n).
- You can invoke the debugger from a program by signaling SS\$_DEBUG.

2.3.3 ANALYZE/RMS_FILE Utility—New Commands Added

The following commands are new:

- NEXT
- BACK
- POSITION/BUCKET
- POSITION/RECORD

These commands make it easier to examine file structures interactively. Also, new integrity features check more thoroughly for file structure errors. Refer to the *VAX/VMS Analyze/RMS File Utility Reference Manual* for further information.

2.3.4 Error Log Utility—New Features and Changes

The new features and changes that have been added to the Error Log Utility are outlined in the following two sections.

2.3.4.1 Enhancements to the User Interface

The ANALYZE/ERROR_LOG command has been enhanced to support the following:

- 1 New device class keywords for /EXCLUDE and /INCLUDE:
WORKSTATION Include or exclude workstation error log entries.
LINE_PRINTER Include or exclude line printer error log entries.
- 2 The BUSES keyword that is supported by /INCLUDE and /EXCLUDE has been enhanced to include BI bus error log entries.
- 3 The DEVICE_ERRORS keyword that is supported by /INCLUDE and /EXCLUDE has been enhanced to include BI adapter error log entries.

The error log entries for workstations and line printers can also be specified by indicating the device name or the type of entry that is logged for the new hardware to /INCLUDE and /EXCLUDE.

2.3.4.2 /EXCLUDE Qualifier Added

By default, whenever an “unknown” device, CPU, or error log entry is encountered by ANALYZE/ERROR_LOG, it will output the entry in a hexadecimal longword format. The /EXCLUDE=UNKNOWN qualifier excludes these entries from the report.

2.3.5 SORT/MERGE Utility—Changes

The following changes relate to the VAX/VMS SORT/MERGE Utility.

2.3.5.1 SORT/MERGE Message Symbols Made Universal

The following SORT/MERGE message symbols were made universal for Version 3.0 compatibility:

```
SOR$_BADLOGIC
SOR$_CLOSEDEL
SOR$_CLOSEIN
SOR$_CLOSEOUT
SOR$_INSVIRMEM
SOR$_OPENIN
SOR$_OPENOUT
SOR$_READERR
SOR$_SYSERROR
SOR$_WRITEERR
```

2.3.6 Print Symbiont—User Defined I/O Routines Supported

A synchronous return of status from a user-defined output routine is supported in Version 4.4. Previously, if a user-supplied output routine returned synchronous status the modified symbiont was not supported for checkpointing. Furthermore, the output of the DCL command SHOW QUEUE displayed the state of the output queue associated with the modified symbiont as "stalled."

In addition, user-defined input filter routines correctly allow the modification of the carriage control of the input record.

User-defined main input routines that do not support the function code PSM\$K_REWIND are automatically called with function code PSM\$K_CLOSE followed by the function code PSM\$K_OPEN. This call sequence provides the intended function of the PSM\$K_REWIND code. This allows user-modified symbionts to support search and alignment operations provided by the standard VAX/VMS print symbiont.

The print symbiont routine PSM\$READ_ITEM_DX is now supported for user-supplied output routines. Previously, PSM\$READ_ITEM_DX was not properly supported when called from a user-defined output routine.

2.3.7 VAX/VMS System Services—New Features Added

Three new system services and a new attribute have been added to the System Services for Version 4.4. The new features are outlined in the following sections. Refer to the *System Services Reference Manual* for detailed information.

New Services Added

There are three new system services in VAX/VMS Version 4.4:

- \$CHECK_ACCESS
- \$GETUAI
- \$SETUAI

New and Changed Features

New Attribute Added

In VAX/VMS Version 4.4, the DYNAMIC attribute is being added to the list of attributes (currently a list of one) found in many of the security-related system services. The following system services are affected:

- \$ADD_HOLDER
- \$ADD_IDENT
- \$ASCTOID
- \$FIND_HELD
- \$FIND_HOLDER
- \$GRANTID
- \$IDTOASC

The format for DYNAMIC is as follows:

Bit	Meaning When Set
KGB\$_DYNAMIC	Allows the unprivileged holder to add or remove the identifier from the process rights list.

2.3.7.1 PRINT USING Function—Behavior Change

The behavior of the PRINT USING function for backslash characters has been changed. In prior versions of VAX/VMS, if a backslash was not followed by another backslash and was not followed by another format sequence, the backslash and other string constants following it were ignored. If it was followed by another format sequence, BAS\$_PRIUSIFOR was signaled.

The behavior is now compatible with BASIC-PLUS. In both cases, the backslash is treated as a string constant (just as if it had an underscore in front of it).

2.3.7.2 SORT/MERGE—/INCLUDE and /OMIT Qualifiers

Previously, SORT/MERGE incorrectly processed conditions specified by the /INCLUDE or /OMIT qualifiers that tested packed decimal fields near the end of a record. The affected record was never included in the output file. In Version 4.4, the record is correctly included or omitted based on the condition specified in the /INCLUDE or /OMIT qualifier.

2.3.7.3 SORT/MERGE—%D Radix Designation Error

In prior versions of VMS, use of the %D radix designation caused an error. It now correctly causes the constant to be treated as a decimal constant.

2.3.8 Run-Time Library—New Features Incorporated

The following sections contain information about new features that have been added to the Run-Time Library. These features include new procedures, new arguments, and a new facility.

A complete description of the new features for the Run-Time Library routines is located in the *VAX/VMS Run-Time Library Routines Reference Manual*.

New and Changed Features

DTK\$, a new Run-Time Library Facility, is available in Version 4.4. For use with DIGITAL's DECTalk devices, it contains all the routines in the following list.

New DTK\$ Procedure	Function
DTK\$ANSWER_PHONE	Answers the phone
DTK\$DIAL_PHONE	Dials the phone
DTK\$HANGUP_PHONE	Hangs up the phone
DTK\$INITIALIZE	Initializes the DECTalk device
DTK\$LOAD_DICTIONARY	Loads a word into the DECTalk dictionary
DTK\$READ_KEYSTROKE	Reads a key entered on the keypad
DTK\$READ_STRING	Reads a series of keys entered on the keypad
DTK\$RETURN_LAST_INDEX	Returns the last index spoken
DTK\$SET_INDEX	Inserts an index
DTK\$SET_KEYPAD_MODE	Turns the phone keypad on and off
DTK\$SET_LOGGING_MODE	Sets or resets logging mode
DTK\$SET_MODE	Sets or resets the specified mode
DTK\$SET_SPEECH_MODE	Starts or stops DECTalk's speech
DTK\$SET_TERMINAL_MODE	Sets or resets terminal mode
DTK\$SET_VOICE	Sets the voice characteristics of the DECTalk device
DTK\$SPEAK_FILE	Speaks the text in the specified file
DTK\$SPEAK_PHONEMIC_TEXT	Speaks the specified phonemic text
DTK\$SPEAK_TEXT	Speaks the specified text
DTK\$TERMINATE	Terminates the use of the DECTalk device

2.3.8.1 New Procedures

There are several new procedures available in Version 4.4. These procedures and their functions are as follows:

New Procedure	Function
New LIB\$ Procedures	
LIB\$PAUSE	Suspends program execution
New SMG\$ Procedures	
SMG\$COPY_VIRTUAL_DISPLAY	Creates a copy of a virtual display
SMG\$DISABLE_BROADCAST_TRAPPING	Disables the trapping of broadcast messages
SMG\$GET_KEYBOARD_ATTRIBUTES	Retrieves information about a virtual keyboard
SMG\$GET_PASTING_INFO	Retrieves information about a virtual display

New and Changed Features

New Procedure	Function
SMG\$REPLACE_INPUT_LINE	Replaces lines in the recall buffer with a specified string
SMG\$RETURN_INPUT_LINE	Returns a line from the recall buffer
SMG\$SET_CURSOR_MODE	Turns the physical cursor on and off

2.3.8.2 New Arguments

New arguments have been added to the following existing routines:

- SMG\$CREATE_VIRTUAL_KEYBOARD
- SMG\$MOVE_VIRTUAL_DISPLAY
- SMG\$PASTE_VIRTUAL_DISPLAY
- SMG\$PUT_LINE
- SMG\$READ_COMPOSED_LINE
- SMG\$READ_KEYSTROKE
- SMG\$READ_STRING
- SMG\$READ_VERIFY
- SMG\$REPASTE_VIRTUAL_DISPLAY
- SMG\$SNAPSHOT

2.3.8.3 Obsolete Routines

The following procedures are now obsolete:

- The Run-Time Library routine LIB\$SYS_TRNLOG is obsolete because it is a jacket routine for the now obsolete system service SYS\$TRNLOG. It is suggested that users directly invoke the new SYS\$TRNLNM system service for logical name translation. Due to the increased capabilities of higher-level languages in constructing item lists, the Run-Time Library has no current plans to provide a jacket routine to this new system service.
- SMG\$PUT_WITH_SCROLL. The routine SMG\$PUT_LINE now supports scrolling, therefore the SMG\$PUT_WITH_SCROLL routine is obsolete.
- SMG\$ALLOW_ESCAPE. This routine was created solely for the purpose of translating old application programs that send escape sequences to SMG\$, and is no longer supported.

2.3.8.4 Other Changes

These routines remain in the documentation for this update but will be removed for the next major release of the documentation.

- The capability for nonminimal updating is now supported. Non-minimal updating redraws only those lines affected by a change, beginning at the first changed character and proceeding to the end of the line.
- Documentation has been added to Chapter 3 concerning user-written exit handlers for screen management routines. This documentation explains why pasteboards and virtual keyboards cannot be deleted from within a user-written exit handler.

New and Changed Features

- The output for LIB\$SHOW_TIMER has changed. Previously, the “elapsed time” output was of the form HHHH:MM:SS.CC. In Version 4.4, this has changed to DDDD HH:MM:SS.CC. Therefore, the output of the example program in the documentation should now read:

```
ELAPSED:  0 00:00:00.22 CPU: 0:00:00.06 BUFI0: 1 DIRI0: 0 FAULTS: 18
```

- The description for the p-kit argument in the SMG\$GET_KEYBOARD_ATTRIBUTES routine should be the same as the pb-info-table argument description in SMG\$GET_PASTEBOARD_ATTRIBUTES.
- As of Version 4.4, RMS provides full record interlocking for shared sequential organization files. Prior to the FT Update (Y4.4), the Run-Time Library would set the FAB\$B_SHR option bit FAB\$V_UPI if the user wished to share a sequential file. In the FT Update, the RTL no longer sets that bit in order to take advantage of full RMS record interlocking for shared sequential files. (This affects programs written in VAX FORTRAN, VAX PASCAL, and VAX PL/I.)

Setting the UFO bit in the file options field requires that UPI also be set. Thus, when you set the FAB\$V_UFO FOP bit in your USEROPEN routine, you must also now set the UPI bit in the SHR field as well. The RTL will no longer set this bit for you.

- The function of LIB\$STAT_TIMER has changed for Version 4.4. The elapsed time returned is now a delta time. Previously, the elapsed time was returned as a difference of two absolute times.
- In a future release of the RTL, users should notice that many parameter names will change. This is in conjunction with an effort to make all RTL parameter names more consistent throughout the various RTL facilities.

For example, in a future release of the SMG\$ RTL, the function-specific flag parameters will all be renamed to a generic parameter named FLAGS. Users using these specific flag parameters should specify a value of 1 to set the flag and 0 to clear the flag. This will allow this parameter to be used in an upward-compatible manner. The following SMG\$ routines will be affected by this parameter name change:

Routine	Parameter
SMG\$CREATE_PASTEBOARD	preserve-screen-flag
SMG\$DELETE_PASTEBOARD	clear-screen-flag
SMG\$PUT_CHARS	erase-flag
SMG\$PUT_LINE	wrap-flag
SMG\$PUT_LINE_WIDE	wrap-flag
SMG\$PUT_LINE_HIGHWIDE	wrap-flag
SMG\$PUT_PASTEBOARD	p-ff-flag
SMG\$SNAPSHOT	ff-flag
SMG\$PUT_WITH_SCROLL	wrap-flag
SMG\$READ_COMPOSED_LINE	function-keys-flag

New and Changed Features

2.3.8.5 Screen Management Restriction

Due to changes made to the Screen Management Facility, the following restriction now applies to the routines `SMG$SET_BROADCAST_TRAPPING`, `SMG$ENABLE_UNSOLICITED_INPUT`, and `SMG$SET_OUT_OF_BAND_ASTS`. For AST routines written in a language that does not support optional parameters (for example VAX BASIC), all system parameters must be specified. This restriction is illustrated in the example for the `SMG$DISABLE_BROADCAST_TRAPPING` routine.

2.3.9 VAX RMS—New Features

Under VAX/VMS Version 4.4, VAX RMS now supports full file sharing, record locking, and the use of global buffers for all sequential files. Also, you may now define descending keys for indexed files. See the *VAX Record Management Services Reference Manual* for details.

2.3.10 Terminal Driver Support—Changes

The following changes have been made to VAX/VMS terminal support.

2.3.10.1 Sending a Break

Previously, there was no \$QIO System Service that allowed an application to send a break to a terminal. A break can now be sent by setting `TT$M_BREAK` in the P5 parity flags argument to the set mode \$QIO.

Sending the break actually involves two \$QIOs; one to turn it on and one to turn it off. (The break bit in the parity flags argument is set to turn on break and cleared to turn it off.) The application should use a timer in between the two \$QIOs to ensure that the break has time to take effect.

2.3.10.2 Preventing Partial Escape Errors

Prior to Version 4.4, the only way to correct partial escape errors (`SS$_PARTEscape`) was for the application program to do single-character reads to parse the remaining characters to determine when the escape sequence was terminated. The terminal driver now supports an alternative approach that allows the application to specify an overflow buffer to be used only for an escape sequence terminator.

A new item code for the item-list read, `TRM$_ESCTRMOVR`, specifies the number of bytes in the read buffer to be reserved for the escape terminator. The P2 parameter, which specifies the size of the read buffer, should include both the number of bytes to receive data and the number of bytes reserved for the escape terminator overflow. Normally the overflow area can be small, perhaps about 10 bytes, since that is sufficient to hold any escape sequence generated by a DIGITAL terminal.

When the terminator overflow area is specified, any bytes from an escape sequence terminator that will not fit in the data area of the buffer will be allowed to occupy the overflow area in the buffer. For instance, a user would be able to type a single character terminated by a keypad key and not get the `SS$_PARTEscape` error, even when the data area is limited to 1 byte.

2.3.10.3 SET HOST/DTE Can Generate a Break

In order to log in on lines that expect a break rather than carriage return, you can now generate a break in SET HOST/DTE by pressing the `CTRL` and Right Bracket (]) keys.

2.3.10.4 SET HOST/DTE/DIAL—Problem and Solution

SET HOST/DTE/DIAL does not work with the DMF-32 controller. The problem is that the modem sends a response character to the host when it detects a carrier signal, but the DMF-32 drops any input until it sees the carrier signal.

One solution is to modify the example autodialer provided in `SYS$EXAMPLES:DT_DF03.MAR` to perform a `IO$_SENSEMODE!!IO$_M_RD_MODEM $QIO` to check for a carrier signal. If set, the autodialer should assume success and continue.

2.3.10.5 Other Changes

In Version 4.0, lines with the MODEM characteristic would hang up 30 seconds after sensing a CARRIER signal if a channel was not assigned to the device. This feature was implemented as a security feature to prevent unused lines from being tied up. It is now possible to disable this hangup on a systemwide basis by setting bit 2 (value = 4) in the `SYSGEN` parameter `TTY_DIALTYP`.

2.3.11 Logical Names Associated With Mailboxes and Mounted Volumes—Changes

In Version 4.4, logical names associated with mailboxes and mounted volumes are no longer automatically deleted unless they are shared logical names. Because all default cases result in the creation of shared logical names, this change is likely to affect only a small number of applications that have deliberately redirected the logical names associated with mailbox creation or the mounting of a volume.

When a mailbox is created, an optional logical name can be associated with the mailbox. Names associated with temporary mailboxes are placed in the logical name table located by the following name:

`LNМ$TEMPORARY_MAILBOX`

Names associated with permanent mailboxes are placed into the table located by the name:

`LNМ$PERMANENT_MAILBOX`

The default assignments for these names are `LNМ$JOB` and `LNМ$SYSTEM` respectively.

When a volume is mounted, a logical name is placed into a table whose name depends on the type of mount operation performed.

Qualifier	Name Table
-----	-----
none	<code>LNМ\$JOB</code>
<code>/GROUP</code>	<code>LNМ\$GROUP_XXXXXX</code>
<code>/SYSTEM</code>	<code>LNМ\$SYSTEM</code>

New and Changed Features

Because all of these logical name tables translate to shared tables, the default behavior applies. When a mailbox disappears (the last process deassigns its channel to the mailbox) or a volume is dismounted, the shared logical name is deleted. The change in Version 4.4 only applies when one of these table names (such as LNM\$TEMPORARY_MAILBOX or LNM\$JOB) is redirected to a process-private table (such as LNM\$PROCESS_TABLE). In the case of redirection, the logical name associated with a mailbox or mounted volume remains when the mailbox disappears or the volume is dismounted.

2.3.12 VAX BASIC—Support Changes

This section explains changes relating to support for VAX BASIC.

2.3.12.1 Error Messages—Modifications and Additions

- The severity of BASIC error 116 (PRIUSIFOR, PRINT-USING format error) has been changed from FATAL to SEVERE. This allows the error to be trapped by a BASIC error handler.
- Two new errors have been introduced: BASIC error 190 (ILLNETOPE, Illegal network operation) and BASIC error 191 (ILLTFFOPE, Illegal terminal-format file operation).
- BASIC error 190 is returned when a program attempts to mix the use of the GET or INPUT statements with PUT or PRINT statements on a terminal format file located on a remote node.
- BASIC error 191 is returned when a program uses the GETRFA built-in function on a terminal format file. Both errors can be avoided by opening the file with ORGANIZATION SEQUENTIAL VARIABLE.
- Previously, when executing on a terminal format file, the GETRFA function exhibited different behavior based on whether the file was remote or local. For a remote file, the function signaled BASIC error 131 (NO_CURREC, No current record). For a local file, the function successfully returned and the GET statement using the RFA returned signaled BASIC error 141 (ILLOPE, Illegal operation). In VAX/VMS Version 4.4, the GETRFA function always returns BASIC error 191 (ILLTFFOPE, Illegal terminal-format file operation).
- In previous releases, mixed usage of GET or INPUT statements with PUT or PRINT on a remote terminal format file caused BASIC error 12 (FATSYSIO, Fatal system I/O failure). This condition now causes BASIC error 190 (ILLNETOPE, Illegal network operation).

2.4 System Programmer Information

This section contains information about the new and changed features of Version 4.4 that are of interest to system programmers.

2.4.1 System Dump Analyzer—New and Changed Features

The following new commands have been added to the System Dump Analyzer:

- ATTACH
- SPAWN

Also, the following new qualifiers are available for the EVALUATE, EXAMINE, and SEARCH commands:

- EVALUATE /PSL
- EVALUATE /PTE
- EVALUATE /SYMBOLS
- EXAMINE /NOSUPPRESS
- EXAMINE /PTE
- SEARCH /LENGTH=length_specifier
- SEARCH /STEPS=step_factor

The following commands are new or changed for Version 4.4:

- The ATTACH command allows you to switch control of your terminal to another process in your job. The /PARENT qualifier allows you to switch control of your terminal to the parent process of the current process.
- The SPAWN command creates a subprocess from the current process. The context is copied from the current process to the spawned process.
- The EVALUATE/PSL command evaluates the specified expression in the format of a processor status longword.
- The EVALUATE/PTE command interprets and displays the expression as a page table entry (PTE). The individual fields of the PTE are separated and an overall description of the PTE's type is provided.
- The EVALUATE/SYMBOLS command specifies that all symbols that are known to be equal to the evaluated expression are to be displayed.
- The EXAMINE/NOSUPPRESS command inhibits the suppression of zeros when displaying memory with one of the following qualifiers: /ALL, /P0, /P1, /SYSTEM.
- The SEARCH/LENGTH command specifies the size of the expression value to be used for successful matching during searches of memory. The possible values of this qualifier are: BYTE, WORD, and LONGWORD.
- The SEARCH/STEPS command controls the granularity of searching through the specified memory range. As each comparison of memory occurs, the value of this qualifier determines the next memory location to be searched. The possible step_factors are: BYTE, WORD, LONGWORD, and QUADWORD.
- The COPY command releases the dump pages in the paging file so that they are available for system paging. Note that once the COPY command has released the dump pages for paging use, the dump information in these pages may be lost. Subsequent dump analysis should be carried out on the copy of the dump file that was specified in the COPY command.

New and Changed Features

- Logical operations have been added to the SDA. They are the logical AND, logical OR, logical XOR. The operators for these operations are &, |, and \.
- The SET PROCESS and SHOW PROCESS commands can now include quoted strings in the process name in addition to the previous capital letters, numbers, dollar sign (\$), and underscore (_) characters.
- The SHOW DEVICE command examples have been changed and now include shadowed devices.
- The SHOW CRASH command register list now includes the system identification register.
- The SHOW PROCESS /RMS=IFAB display has been altered to show the changes to that display.

For more information, refer to the *VAX/VMS System Dump Analyzer Reference Manual*.

2.4.2 CSMA/CD Data Link Drivers—New Features

The CSMA/CD data link (XE and XQ) drivers support the following elements of the IEEE 802 standard: the 802.2 and 802.3 packet format; 802.2 Class I service; 6-byte destination and source address fields; and a physical layer identified as type 10BASE5 (10Mb/s baseband medium with maximum segment length of 500 m).

2.4.3 TS11/RSX-11—Operation Restriction

TSDRIVER.EXE—The BRU (Backup and Restore Utility) under the RSX-11 emulator will not operate when the target device is a TS11 on multi-volume operations.

2.4.4 DR11-W/DRV11-WA (XADRIVER)—New Support

The DR11-W is a VAX/VMS supported 16-bit parallel direct memory access (DMA) interface for UNIBUS systems. VAX/VMS includes the source code for the DR11-W driver so that users can tailor the driver for their own applications.

Beginning with Version 4.4, VAX/VMS and MicroVMS also support the DRV11-WA, a 16-bit parallel DMA interface on the Q-bus. Because the DRV11-WA and DR11-W interfaces are similar and many users of the DR11-W wish to convert their applications to run with the DRV11-WA, support for the DRV11-WA interface has been folded into the DR11-W driver.

The Version 4.4 XADRIVER does not contain bug fixes or enhancements that affect the DR11-W interface. If a customer has not tailored the driver (SYS\$SYSTEM:XADRIVER.EXE), the new version of the driver will function on the DR11-W as the old. However, if any changes have been made to the driver, it is suggested those changes be merged into the Version 4.4 driver.

The Version 4.4 XADRIVER documentation and source code presume the DRV11-WA is at CS Revision Level B and Etch Revision Level D or earlier. If subsequent revisions are made to the board, the expected behavior of the driver is unpredictable.

For additional information, see *VAX/VMS I/O User's Reference Manual: Part II* and the *DRV11-WA General Purpose DMA Interface User's Guide*.

2.4.5 CI Port Driver (PADRIVER)—New Image

The following is a brief summary of the Version 4.4 changes in the computer interconnect (CI) port driver, PADRIVER:

2.4.5.1 Supported Microcode

For proper PADRIVER support, all sites should have at least Version 5.0 of the CI780 microcode. Version 5.0 microcode has known problems that produce CI port shutdowns for a variety of reasons. These shutdowns occur most frequently on large, heavily loaded clusters. If these shutdowns adversely affect the stability of your cluster, you should inquire about the availability of the next release of CI microcode, Revision 7.0.

Version 7 of the CI microcode will contain a variable sanity timer. The presence of this timer will enable the system communication services (SCS) virtual circuit timeout mechanism that is described in this release note.

The current microcode version can be identified by executing the SHOW CLUSTER /CONTINUOUS DCL command and then typing the ADD RP_REVIS subcommand. The low-order word is the RAM version and the high-order word is the PROM version. For Version 7.0 microcode, this field will contain the value 70007 (hexadecimal).

The port driver will display the following message for sites containing old versions of the microcode:

```
%PAA0, - CI port ucode not at current rev level. PROM/RAM rev is 0004/0003
```

2.4.5.2 SCS Virtual Circuit Timeouts

The VMS Version 4.4 CI port driver and the Version 7.0 CI microcode implement an SCS virtual circuit timeout. This mechanism reduces cluster transition times by allowing rapid detection of a failed cluster node. In VAX/VMS Version 4.3, certain catastrophic hardware failures prevented orderly shutdown of the failing node. Because such failures did not shut down the CI port, other cluster members did not recognize a hardware failure for at least 100 seconds. This period is the expiration time for the sanity timer of the CI port.

The SCS virtual circuit timeout reduces the cluster transition time by requiring that CPUs, not ports, exchange periodic SCS control messages. The failure of a node to respond to an SCS control message within a specified time causes the port driver to break the virtual circuit and notify the connection manager of a communication problem.

With this timeout mechanism enabled, the CI port driver will periodically check to see that it is receiving systems-level messages from all remote VMS processors. The presence of these messages guarantees that the remote processor is not halted or hung in a loop at hardware IPL 7 or greater. Should no routine messages appear from a remote node, the CI port driver will attempt to generate traffic by sending a dummy keep-alive message to the remote.

New and Changed Features

2.4.5.3 Virtual Circuit Timeout Errors

A timeout of the keep-alive message is fatal to the logical link between the two systems. When a timeout occurs, the driver will close the logical link, create an error log entry, and print the following message on the operator's console:

```
%PAA0, Virtual Circuit Timeout - REMOTE PORT nn
```

These messages do not represent a hardware failure but do notify the system manager that the remote node is either halted or spending a long time at high hardware priority levels. Occurrence of the latter depends on the setting of the SYSGEN parameters, the nature of the processing load on the cluster, and finally, on the presence of user-written privileged code. The system manager should first increase the PASTIMOUT timeout parameter (described below) until virtual circuit timeouts occur infrequently if at all. The system manager may then wish to consult the *Guide to VAX/VMS Performance Management* to investigate the general performance characteristics of his system.

2.4.5.4 SYSGEN Parameters

The following SYSGEN parameters affect the SCS virtual circuit timeout mechanism. The system manager must ensure that these parameters are the same on all nodes in a cluster.

PASANITY	Boolean parameter which, when set to zero, disables the timeout mechanism. This parameter is used for the debugging of system code and should not be changed. (Default is 1.) Customers writing privileged code using the XDELTA debugging tool must set PASANITY to zero on the entire cluster to avoid a CLUEXIT bugcheck.
PASTIMOUT	The activation time for the port timeout mechanism of the port driver. The port driver is able to detect a virtual circuit timeout within a minimum of PASTIMOUT seconds and a maximum of 3K PASTIMOUT seconds. The driver will adjust the effective value of PASTIMOUT if the PAPOLLINTERVAL parameter (described below) is too low. The effective value of PASTIMOUT is computed as follows (assuming PAPOLLINTERVAL is less than or equal to PASTIMOUT): Effective PASTIMOUT = MAX(PASTIMOUT, 2*PAPOLLINTERVAL) Do not set PAPOLLINTERVAL greater than PASTIMOUT. Such a setting has no useful purpose. (Default is 5 seconds.)

New and Changed Features

PAPOLLINTERVAL This parameter specifies the amount of time the CI port driver requests the port to poll for other nodes in a cluster. The failure of the port to complete a poll during this interval causes the driver to declare a CI port timeout and reset the port.

The port driver must guarantee detection of a failed local port before detection of failed links to remote nodes. Otherwise, a port failure could result in multiple "virtual circuit timeout" messages for every remote node. The port driver uses the above PASTIMOUT formula to ensure that a port timeout precedes virtual circuit timeouts. (Default is 5 seconds.)

RECNXINTERVAL This parameter specifies the amount of time that the connection manager waits between the loss of a connection to a remote node and the initiation of a cluster transition to remove the failed node from the cluster. The minimum value of RECNXINTERVAL must guarantee that all of the connections seen from the viewpoint of the remote node are broken. Otherwise, the remote node may continue to access cluster disks after being removed from the cluster. **The correct setting of RECNXINTERVAL is at least three times the effective value of PASTIMOUT as determined by the following formula (assuming PAPOLLINTERVAL is less than or equal to PASTIMOUT):**

$$\text{RECNXINTERVAL} > 3 * (\text{MAX}(\text{PASTIMOUT}, 2 * \text{PAPOLLINTERVAL}))$$

Since the default intervals for PASTIMOUT and PAPOLLINTERVAL are 5 seconds, the minimum allowable RECNXINTERVAL is 30 seconds. The default is 60 seconds.

For clusters requiring rapid failover, the system manager can decrease both PASTIMOUT and PAPOLLINTERVAL to 1 second. On heavily loaded clusters, however, this rapid failover may lead to an increase of CLUEXIT bugchecks on individual nodes. Clusters with Version 5.0 CI microcode should retain the default settings, since the port driver does not enable the virtual circuit sanity timer.

System Communication Timeout SYSGEN parameters

Parameter	Default	Minimum
PASANITY	1	1
PAPOLLINTERVAL	5 sec	1 sec
PASTIMOUT	5 sec	1 sec
RECNXINTERVAL	60 sec	6 sec

Result	Default	Minimum
Port failure detection	5-10 sec	1-2 sec
Virtual Circuit failure detection	10-30 sec	2-6 sec

3 **Problems, Restrictions, and Notes**

This chapter contains information pertaining to problems and restrictions of the VAX/VMS Version 4.4 Release. It also includes general notes and documentation notes about the Version 4.4 release. Each topic is given a brief description and a reference to where more information can be found (when applicable).

This chapter is arranged into the categories:

- Section 3.1—General User Information
- Section 3.2—System Manager Information
- Section 3.3—Application Programmer Information
- Section 3.4—System Programmer Information

To find a specific topic, consult the index in the back of this manual.

3.1 **General User Information**

This section contains information about the VAX/VMS Version 4.4 release that pertains to all system users.

3.1.1 **VAX/VMS Document Set—Reorganization**

The VAX/VMS document set has been reorganized significantly for Version 4.4. The documentation update kit, which is sent to existing customers and includes the changes and additions to the document set, includes a cover letter that explains the new organization. The cover letter also explains how to update your current document set to a Version 4.4 document set.

3.1.2 **Version 4.0 Release Notes Appendixes—Disposition of Material**

Appendixes B through G of the *VAX/VMS Release Notes, Version 4.0*, contained information from the Version 3 documentation set that had not been integrated into the Version 4 documentation set. The following table lists the topics and their disposition as reflected in the Version 4.4 documentation.

Problems, Restrictions, and Notes

Topic	Disposition
Introduction to VAX/VMS Input /Output	Integrated into I/O discussion in the <i>VAX/VMS System Services Reference Manual</i>
Real-Time I/O (Mapping I/O Space, Connecting to Interrupt Vector)	Included in the <i>Writing a Device Driver for VAX/VMS</i> manual, Appendix H
User-Written System Services (Privileged Shareable Images)	Included in the <i>VAX/VMS System Services Reference Manual</i> , Appendix B
Using Shared Memory	Included in the <i>VAX/VMS System Services Reference Manual</i> , Appendix C
Obsolete Run-Time Library Routines	No longer documented
Summary of Run-Time Library Entry Points	Use the RTL section of the <i>VAX/VMS Mini-Reference</i>

3.1.3 Mini-Reference—Supersedes Quick-Reference Booklets

The *VAX/VMS Mini-Reference*, a new manual for Version 4.4, contains comprehensive quick-reference information. It supersedes the following previously published quick-reference booklets:

- *VAX/VMS DCL Commands and Lexical Functions*
- *VAX EDT Quick Reference Guide*
- *VAX DSR Quick Reference Guide*
- *VAX/VMS System Services and Run-Time Library Routines*

3.1.4 Terminal Driver Line Editing—Clarification

The following information clarifies the documentation for CTRL/V in Table 1-2 of the *VAX/VMS DCL Concepts Manual* and in Table GEN-2 of the *VAX/VMS Mini-Reference*.

At DCL level, CTRL/V turns off line-editing features that were new with Version 4.0. For example, if you type CTRL/V followed by a control key such as CTRL/D, a CTRL/D is generated instead of the cursor moving left one character. Note, however, that CTRL/D is a terminator at the DCL level. Thus, when you type CTRL/V followed by CTRL/D, a carriage return is simulated. DCL uses the default RMS terminator set. The characters that are part of this set are described in Chapter 8 of the *VAX/VMS I/O User's Reference Manual: Part I*.

When combined with CTRL/V, characters that are not terminators to DCL will have no effect since a backspace or linefeed in the middle of a line would result in an invalid command. Examples are CTRL/H and CTRL/J.

Certain control keys perform the same function with Version 4.0 as they did in previous versions of VAX/VMS. If you type one of these keys (including CTRL/U) after a CTRL/V, the key will behave as it did prior to Version 4.0.

3.1.5 **VAX/VMS Backup Utility Reference Manual—Guide For New User's Added**

The Description section of the *VAX/VMS Backup Utility Reference Manual* now includes a subsection called "Using BACKUP: A Guide for New Users." This section is an introduction to the Backup Utility for inexperienced users; it includes descriptions of BACKUP function, operation types and modes, qualifiers, and save sets. Users who are new to BACKUP, or who use it infrequently, may benefit by reading the guide first and then reading the reference sections of the manual.

3.1.6 **Mail Utility—Restriction in Cluster Running Mixed Versions**

If a user creates a MAIL.MAI file under Version 4.4 and someone attempts to send MAIL to that user on a cluster node that is running a previous version of VAX/VMS, the sender will receive the following error message:

```
%MAIL-E-SENDERR, error sending to user 'username'
-RMS-F-DUP, duplicate key detected (DUP not set)
```

The sender should resubmit the mail message to a node on the cluster that is running Version 4.4.

3.1.7 **VAX/VMS Mail Utility Reference Manual—Text Addition**

The description of the /SELF qualifier in the *VAX/VMS Mail Utility Reference Manual* should include the following information:

- The /SELF qualifier is negatable.
- If you send a message from the DCL level (that is, you do not receive the MAIL> prompt from within the Mail Utility), specifying /SELF or /NOSELF overrides any setting you have established by the SET COPY_SELF command within the Mail Utility.
- Specifying /SELF or /NOSELF on the DCL command line has no effect if you enter the Mail Utility and receive the MAIL> prompt.

Thus, for example, you could specify the following command to send MYFILE.DAT to user RUSCIO, and avoid receiving a copy of the file yourself even if you have previously entered the SET COPY_SELF command within the Mail Utility.

```
‡ MAIL/NOSELF MYFILE.DAT RUSCIO
```

3.1.8 **Guide to Using DCL and Command Procedures on VAX/VMS—Documentation Changes**

The following corrections apply to the *Guide to Using DCL and Command Procedures on VAX/VMS*.

- Page 1–11. The last two lines in the example LOGICALS.COM file should read as follows:

```
‡ DEFINE JON DAISY::HARRIS
‡ DEFINE JANE DAISY::MOORE
```

Problems, Restrictions, and Notes

- Page 4–16. In the example in Section 4.6.2, the statement

```
‡ WRITE "Result is",RES
```

should be replaced as follows:

```
‡ WRITE SYS$OUTPUT "Result is ",RES
```
- Page 5–14. A line of code is missing from the example at the top of the page. The line

```
‡     NUM = NUM + 1
```

should be inserted under PROCESS_LOOP as follows:

```
‡ PROCESS_LOOP:  
‡   FILE = F$ELEMENT(NUM,"/",FILE_LIST)  
‡   IF FILE .EQS. "/" THEN GOTO DONE  
‡   COPY 'FILE'.MEM MORRIS::DISK3:[DOCSET]*.*  
‡   NUM = NUM + 1  
‡   GOTO PROCESS_LOOP
```
- Page 5–15. The first statement in the example should have a hyphen (-) at the end of the line as follows:

```
‡ COMMAND_LIST = "DELETE/DIRECTORY/EXIT/" + -
```
- Page 6–7. In the example at the top of the page, the statement

```
‡ INQUIRE RECORD "Enter name"
```

should be replaced as follows:

```
‡ INQUIRE NAME "Enter name"
```
- Page 8–5. In the text following the third bullet, the qualifier /NOPRINT should be /NOPRINTER.

3.1.9 Extended File Names/Types—Caution

Although file names and file types of up to 39 characters are permitted starting with VAX/VMS Version 4.0, for some files you may need to use the VAX/VMS Version 3.x maximum lengths (9 characters for the file name and 3 characters for the file type), or other maximum lengths as appropriate.

For example, you must use restraint in naming files that will be accessed by:

- Operating systems that cannot support longer file names and file types, such as VAX/VMS Version 3.x systems and systems for PDP-11 processors
- Applications software that will not accept longer file names and file types

Care should be taken when naming files that will be copied or accessed by remote systems. The file-naming abilities of VAX/VMS after Version 4.0 exceed those of most other computer systems, including VAX systems running VAX/VMS Version 3.x. For example, a system running VAX/VMS Version 3.x will return a syntax error when a file specification contains a file name (including a directory name) longer than 9 characters, a file type longer than 3 characters, a dollar sign (\$) or an underscore (_). Valid file specifications of VAX/VMS after Version 4.0 that are invalid on a VAX/VMS Version 3.x system include the following:

```
NODE::DBA2:[YOUR_DIR]FILE.DAT  
NODE::DBA2:[DIR]FILETOOLONG.DAT  
NODE::DBA2:[DIR]FILE_TEST.DAT  
NODE::DBA2:[DIR]FILE.DATA
```

A user of a Version 4.0, or later, system would have to rename (or copy) these files before the remote system could access them. Alternatively, the user could copy these files to the remote system by using valid VAX/VMS Version 3.0 output file specifications.

File name restrictions are generally determined by the file name capabilities of the remote system(s) that require access to the file. Such restrictions should be considered as part of the overall application design when network access is required.

Applications that parse file specifications using the pre-Version 4.0 file specification conventions should be modified to use the services or routines that can parse or scan file specifications using the new extended file specifications conventions. These services and routines include the RMS Parse service and the Scan String for File Specification system service (see the *VAX Record Management Services Reference Manual* and the *VAX/VMS System Services Reference Manual*) and the LIB\$FIND_FILE and LIB\$FILE_SCAN routines (see the *VAX/VMS Run-Time Library Routines Reference Manual*).

3.1.10 Shutdown Notification on Clusters—Note

When the REMOVE_NODE option is specified during execution of an orderly shutdown procedure (SYS\$SYSTEM:SHUTDOWN.COM) on one VAXcluster member system, users on all member systems are notified. Clusterwide notification is required, because users logged in to any member system may be affected by the shutdown of another system in various ways:

- Users may have batch jobs running on other systems.
- If terminal servers are in operation, users may have alternate terminal sessions in progress (for example an editing session) on the system being shut down.

Since shutdown messages include the name of the member system being shut down, users need only check the messages carefully to avoid logging out of a system unnecessarily.

Note that, for those reasons, clusterwide notification is not affected by the shutdown procedure's REPLY/NODE= option. If, for some reason, you wish to limit shutdown notification to specific member systems, define the logical name SHUTDOWN\$INFORM_NODES before executing the shutdown procedure. For example:

```
⚡ DEFINE SHUTDOWN$INFORM_NODES MOE,LARRY
⚡ @SYS$SYSTEM:SHUTDOWN
```

In this example, only users on systems MOE and LARRY will be notified.

3.2 System Manager Information

This section contains information about the VAX/VMS Version 4.4 release pertaining to system managers.

Problems, Restrictions, and Notes

3.2.1 **STAND-ALONE BACKUP—Mandatory Rebuild**

You must rebuild stand-alone BACKUP after you have upgraded or installed Version 4.4 of VMS. It is recommended as normal procedure. DIGITAL wishes to draw special attention to this requirement. Please refer to the *Guide to VAX/VMS System Management and Daily Operations*, for a complete description of how to rebuild stand-alone BACKUP.

3.2.2 **Guide to Multiprocessing on VAX/VMS—Setting Up a VAX-11/782 System**

This section supplements the *Guide to Multiprocessing on VAX/VMS* and provides instructions for building multiprocessing console diskettes on a VAX-11/782 system. The information in this section assumes the following:

- The VAX-11/782 hardware has already been installed and configured.
- The VAX/VMS operating system has been installed as described in the installation booklet packaged with the media (for example, *Installing VAX/VMS on a VAX-11/780 From Magnetic Tape*).

Note that you should keep a record of the following information regarding memory configuration:

- The number and type of memory controllers
- The transmit request (TR) levels at which the controllers are configured
- The amount of memory on each controller

Note: The command procedure `BOOTBLDR.COM` does not recognize MS780-H memory. If your system configuration includes an MS780-H, contact your DIGITAL Customer Support Center or submit an SPR.

3.2.2.1 **Building Multiprocessing Console Diskettes**

Each processor in the VAX-11/782 system must have its own console diskette. Multiprocessing console diskettes allow for the booting of the VAX-11/782 attached processor system by means of several boot command procedures. These bootstrap command procedures cause MA780 shared memory rather than local memory to be used as main memory, and they set the memory configuration registers to ensure that MA780 shared memory is configured at the low physical addresses (beginning at 0) and local memory at the higher addresses.

In addition, each multiprocessing console diskette contains the “reset memory” command procedure `RMEM.COM`, which is specific to the memory configuration of the processor. The `RMEM.COM` procedure reconfigures local memory to start at physical address 0 (zero) and MA780 shared memory to start at adjacent higher physical addresses. Thus, after executing `RMEM.COM`, you can boot the VAX-11/782 system as a single-processor VAX-11/780 system by using a standard VAX-11/780 console diskette (providing you have sufficient local memory on the system).

To build multiprocessing console diskettes, you execute the interactive command procedure `SYS$UPDATE:BOOTBLDR.COM`. This command procedure first creates a new console diskette for the primary processor and then one for the attached processor. The procedure executes interactively and prompts you for information about the memory configuration of the system.

Problems, Restrictions, and Notes

The following sections describe how to obtain information about the memory configuration of the system and how to execute `BOOTBLDR.COM`.

Determining the Memory Configuration

In order to run `BOOTBLDR.COM`, you must determine certain information about the configuration of your system. For each memory controller on the system, you need to know the following:

- Its type (MS780-A, MS780-C, MS780-E, or MA780)
- The transmit request (TR) level at which it is configured
- The amount of memory it holds

In addition, you need to know the TR level of the first UNIBUS and MASSBUS adapter on the system (that is, the UBA and the MBA with the lowest TR number). You will need the information regarding MS780-x memory and MA780 memory for both the primary and attached processors; information regarding UNIBUS and MASSBUS adapters is needed only for the primary processor. (Note that not all VAX 11/782 systems are configured with a MASSBUS adapter.)

You can obtain the necessary information from your DIGITAL Field Service Representative or by following the procedure in this section. If you already know the memory configuration of your system, proceed to the section entitled `Executing BOOTBLDR.COM`.

This section describes how to obtain this information first for the primary processor and then for the attached processor. The procedure is the same in both cases, with the following exceptions:

- For the primary processor, perform the procedure at the primary processor's console terminal; for the attached processor, at the attached processor's console terminal.
- You need to determine the TR level and memory amount of each MA780 controller only once (for the primary processor), since an MA780 memory controller must be configured at the same TR level on both processors. Note however that obtaining this information twice (for both processors) allows you to check that MA780 memory has been configured correctly (is at the same TR level) on both processors.

You can determine the memory configuration of each processor by examining the configuration registers for TR levels 1 through 15. Memory controllers may be configured only at TR levels 1 through 6; UNIBUS and MASSBUS adapters may be configured at any TR level (1 through 15). TR level 0 is reserved for the CPU and is not of interest to `BOOTBLDR.COM`.

Table 3-1 shows the physical addresses for the configuration registers at each TR level. Table 3-2 shows the codes for each type of adapter.

Problems, Restrictions, and Notes

Table 3–1 Configuration Register Physical Addresses

Transmit Request (TR) level	Configuration Register (CR) Physical Address	MA780 Port Invalidation Configuration Register (PICR) Physical Address
1	20002000	2000200C
2	20004000	2000400C
3	20006000	2000600C
4	20008000	2000800C
5	2000A000	2000A00C
6	2000C000	2000C00C
7	2000E000	
8	20010000	
9	20012000	
10	20014000	
11	20016000	
12	20018000	
13	2001A000	
14	2001C000	
15	2001E000	

Table 3–2 Adapter Type Codes

Value	Adapter Type	Scale Factor
08,09	MS780-A	64KB
10,11	MS780-C	64KB
20	MBA (MASSBUS adapter)	n/a
28-2B	UBA (UNIBUS adapter)	n/a
40-43	MA780	256KB
68-6A	MS780-E	1024KB
70-74	MS780-H	n/a
Other	not of interest to BOOTBLDR.COM	

To determine the memory configuration on your system, perform the following steps:

- 1 Put the console terminal into console mode by typing CTRL/P.
- 2 Enter the console command HALT at the console prompt (> > >).
- 3 Use the console command EXAMINE to read the appropriate configuration registers for each TR level (Table 3–1 shows the TR levels and their corresponding registers). For example, the following command reads the configuration register at TR level 1 (TR1):

```
>>>EXAMINE 20002000
P 20002000 00002610
```


Problems, Restrictions, and Notes

If the specified TR level has no adapter at all, you will see a display like the following, where *n* represents a hexadecimal digit:

```
>>>EXAMINE 2000E000
? MIC-ERR ON FUNCTION
(nn)nnnnnnnn (nn)nnnnnnnn (nn)nnnnnnnn
(nn)nnnnnnnn (nn)nnnnnnnn (nn)nnnnnnnn
(nn)nnnnnnnn (nn)nnnnnnnn (nn)nnnnnnnn
```

The EXAMINE command in this example attempts to read the configuration register for TR level 7. Because there is no adapter at TR 7, the microcode returns an error after trying to read the nonexistent CR.

- 4 Using Table 3-2 for reference, interpret the displayed value to determine the type of adaptor connected to the designated TR level. To do this, extract the two rightmost digits from the configuration register (CR) display and match them with an entry listed under "Value" in the table. (Note that the table only shows the values of interest to BOOTBLDR.COM).

For example, the CR value displayed by the EXAMINE command in step 3 is 00002610. The two rightmost digits are 10. According to Table 3-2, the value 10 designates an MS780-C memory controller containing 64-kilobyte memory boards.

- 5 Depending upon the type of adapter (MS780, MA780, UNIBUS or MASSBUS), perform the following steps:
 - MS780 memory (all types except MS780-H)
 - a Extract the fifth and sixth (from the left) digits from the CR value. (In the example above, these digits are 26.)
 - b Convert this number from hexadecimal to decimal.
 - c Divide the converted number by 2, discarding the remainder, and add 1 to the result.
 - d Multiply the result by the scale factor (shown in Table 3-2) to determine the total amount of memory.
 - e Record the TR number, memory type, and the amount of memory for later use.

Note that the BOOTBLDR.COM procedure does not recognize MS780-H memory. If your configuration includes this type of memory, contact your DIGITAL Customer Support Center or submit an SPR.

- MA780 memory
 - a Examine the contents of the MA780 Port Invalidation Configuration Register (PICR).
 - b Extract the fourth digit from the left and add 1 to that value. (There is no need to convert to decimal, as the value is never greater than 7.)
 - c Multiply the result by the scale factor (shown in Table 3-2) to determine the total amount of memory.
 - d Record the TR number, memory type, and the amount of memory for later use.
- UNIBUS or MASSBUS adapters (UBA or MBA)

Problems, Restrictions, and Notes

Make note of the TR number of the first (lowest TR number) of each type.

A memory controller may be configured only at TR levels 1 through 6. However, a UBA or MBA may be configured at any TR level from 1 through 15. Therefore, you should examine the locations that correspond to TR levels 7 through 15 to determine whether a UBA or MBA is configured at any of them. If the two rightmost digits of the displayed value are in the range 28 through 2B, a UBA is configured at the TR level corresponding to the examined address. If the two rightmost digits of the displayed value are 20, an MBA is configured at the TR level corresponding to the examined address.

Note that if the rightmost two digits of the value displayed by any EXAMINE command are not in the ranges shown in Table 3-2, the TR level corresponding to the examined address does not have a memory controller, UBA, or MBA. In this case, whatever is configured at that particular TR level is not of concern, and you need not further interpret the displayed value. Further, if a microcode error results when you examine any of the addresses, simply assume that a device is not configured at the TR level corresponding to the examined address.

- 6 Once you have completed the above steps, you have determined the memory configuration for the primary processor. To determine the memory configuration of the attached processor, repeat steps 1 through 4 at the attached processor's console terminal. That is, put the terminal in console mode, issue the EXAMINE commands for TR levels 1 through 6, and interpret the displayed values.

Again, you need not obtain information about MA780 memory a second time, since information about MA780 memory is identical for both the primary and attached processors. Thus, you need not examine one of the six addresses if an examination of that address at the primary processor's console terminal revealed an MA780 memory controller. However, it may be useful to examine all addresses in order to verify that your system is configured properly.

Once you have completed steps 1 through 5, you have determined the memory configuration of the system. You have all the information needed to execute BOOTBLDR.COM. Proceed to the next section.

Executing BOOTBLDR.COM

BOOTBLDR.COM is an interactive command procedure that builds console diskettes for the primary and attached processors. You must execute BOOTBLDR.COM when you initially set up your VAX-11/782 system and whenever you change its memory configuration. The multiprocessing console diskettes created by BOOTBLDR.COM can only be used in a system with the same memory configuration as the memory configuration of that system for which they were created.

BOOTBLDR.COM requests that you enter information, and it gives you instructions about what to do next. As it executes, it displays messages that indicate what is taking place.

This section discusses those parts of the command procedure over which you have control. That is, it discusses how to respond to requests for information. If you want more information, you can read the command procedure itself by entering the following command at the console terminal:

```
‡ TYPE SYS$UPDATE:BOOTBLDR.COM
```

Note: The console diskettes you are creating initialize the starting physical addresses of all memory on the system. If you enter incorrect memory amounts when executing BOOTBLDR.COM, these starting physical addresses will be incorrect. A machine check will result if VAX/VMS references an incorrect physical address.

The following steps describe how to use BOOTBLDR.COM.

- 1 Enter the following command to invoke the procedure:

```
$ @SYS$UPDATE:BOOTBLDR
```

The procedure prompts as follows:

```
Enter memory type (MA780, MS780C, MS780E or <RETURN> to end):
```

- 2 Enter the name of the first memory controller that is configured on the primary processor. For example, enter MS780E if you have an MS780E memory controller configured on the primary processor. Do not abbreviate or add suffixes to your response; for example, do not abbreviate MS780E as MS.

There are no defaults; do not press RETURN until you have entered all the necessary information (the procedure will continue to prompt for memory type until you press RETURN).

The procedure then prompts as follows:

```
Enter TR level (1 through 6):
```

- 3 Enter the number of the TR level at which the memory controller (MS780 or MA780) you entered in response to the previous prompt is configured. Enter only a number. Note that if you simply pressed the RETURN key in response to the previous question, this prompt does not appear.

The procedure then prompts as follows:

```
Enter amount of memory for this controller in .25 megabyte increments (for example, for 512 kilobytes, enter .5):
```

- 4 Enter the amount of memory configured at this TR level. Enter only a number; that is, do not enter a suffix such as Mbytes or megabytes. Note that memory must be present in increments of .25 megabytes.

The procedure repeats this sequence of requests until you press RETURN (and nothing else) in response to the first request. You should press RETURN after you have named all memory controllers connected to the primary processor.

After you respond to a prompt by pressing RETURN, the procedure displays the following message:

```
Would you like the bootstrap command files to boot the system using local (MS780A, MS780C, or MS780E) memory as well as shared (MA780) memory <YES or [NO]>:
```

- 5 Press RETURN. The procedure responds with the following information and prompt:

```
The UNIBUS Adapter (UBA) is assumed to be at TR level x.  
Enter the TR level of the UBA (Enter <RETURN> to default):
```

The letter x represents a number from 1 through 15. BOOTBLDR.COM derives the number represented by the letter x by adding 1 to the number of the highest TR level at which an MA780 memory controller is configured. BOOTBLDR.COM uses the TR level of the UBA in the

Problems, Restrictions, and Notes

creation of boot command procedures (such as DM0BOO.COM) for UNIBUS devices (such as RK06 and RK07 disk drives).

If a UBA is configured at the TR level displayed in the preceding message, simply press RETURN; do not enter a number. On the other hand, if a UBA is not configured at the TR level displayed in the preceding message, enter the TR level at which the UBA is configured; enter only a number and then press RETURN.

Note that a system may have more than one UBA and that BOOTBLDR.COM can create boot command procedures for use on only one UBA. In a system with more than one UBA, you must select the UBA for which you want boot command procedures created. In this way, boot command procedures for devices on that UBA (but not for devices on the other UBA) will be created.

- 6 The procedure continues with a similar prompt for the MASSBUS adapter: and prompt:

The Massbus Adapter (MBA) is assumed to be at TR level x.
Enter the TR level of the MBA (Enter <RETURN> to default):

If an MBA is configured at the TR level displayed in the preceding message, simply press RETURN; do not enter a number. If an MBA is not configured at the TR level displayed in the preceding message, enter the TR level at which the MBA is configured; enter only a number and then press RETURN.

The procedure continues by prompting as follows:

Enter the name of the default boot command procedure (DEFBOO.COM)
to be used when booting the system. (Default is xxnBOO.COM):

VAX/VMS supplies a number of default boot command procedures to enable you to boot the system from various devices. In general, the file name of the default boot command procedure you should choose has as its first three characters the device name of the device on which you expect the system disk to reside; the remaining characters in the file name are BOO.COM. Respond to the request by entering the file name (for example, DB0BOO.COM).

- 7 Next, the procedure asks for information about the memory configuration for the attached processor. Before prompting you for the information, BOOTBLDR.COM reminds you that MA780 memory must be identical on both processors. For this reason, the procedure will not prompt for the TR level(s) at which MA780 memory is configured on the attached processor. You have already provided the necessary information about MA780 memory.

The procedure then mentions that local (MS780A, MS780C, or MS780e) memory on the attached processor may be different from local memory on the primary processor. That is, MS780 memory on the attached processor may be configured at different TR levels; further, there may be more or less MS780 memory on the attached processor.

The procedure then prompts as follows:

Enter memory type (MA780, MS780C, MS780E or <RETURN> to end):

- 8 Enter the name of the first memory controller that is configured on the attached processor.

The procedure then prompts as follows:

Enter TR level (1 through 6):

Problems, Restrictions, and Notes

- 9** Enter the number of the TR level at which the memory controller (MS780 or MA780) you entered in response to the previous prompt is configured.

The procedure then prompts as follows:

```
Enter amount of memory for this controller in .25 megabyte
increments (for example, for 512 kilobytes, enter .5):
```

- 10** Enter the amount of memory configured at this TR level.

The procedure repeats this sequence of requests until you press RETURN (and nothing else) in response to the first request. You should press RETURN only after you have named all memory controllers connected to the attached processor.

- 11** After you respond to a prompt by pressing RETURN, the procedure prompts you for the name of the diskette drive you want to use. Enter the drive name as in the following example. BOOTBLDR.COM then instructs you to insert the original diskette in the drive and asks whether you are ready to continue.

```
Enter the name of the floppy disk drive you want to use: CSA1
Insert original 11/780 console floppy in CSA1:.
Ready to continue? (YES or NO): YES
Copying console floppy to temporary directory.
Copying VMB.EXE from SYS$SYSTEM.
11/782 requires a V3 or later VMB in order to use MA780 memory.
```

- 12** Next, BOOTBLDR.COM instructs you to remove the original diskette and to place a scratch volume in the drive.

```
Please remove original floppy from CSA1:
Creating floppy for primary processor.
Place a scratch floppy in CSA1:.
WARNING -- CSA1: will be initialized.
```

After warning you that the volume will be initialized, the procedure asks whether you are ready to continue. Enter YES (assuming you have placed a scratch diskette in the drive as instructed). BOOTBLDR.COM proceeds as follows:

```
Ready to continue? (YES or NO): YES
Note: Console media must not contain any bad blocks.
Analyzing CSA1: for defective blocks, please stand by...
%MOUNT-I-MOUNTED ...
Copying unmodified files to CSA1:.
Creating multiprocessor bootstrap command procedures.
Primary processor console floppy completed.
Creating floppy for attached processor.
Place a scratch floppy in CSA1:.
WARNING -- CSA1: will be initialized.
Ready to continue? (YES or NO):
```

If BOOTBLDR.COM finds any bad blocks, it will not mount the volume. Instead, it asks you to place another scratch volume in the drive.

- 13** Remove the first scratch volume and place another scratch volume in the drive. Enter YES to continue; BOOTBLDR.COM proceeds as follows and completes the build.

```
Ready to continue? (YES or NO): YES
Note: Console media must not contain any bad blocks.
Analyzing CSA1: for defective blocks, please stand by...
%MOUNT-I-MOUNTED ...
Copying unmodified files to CSA1:.
Creating multiprocessor bootstrap command procedures.
Attached processor console floppy completed.
```

Problems, Restrictions, and Notes

If BOOTBLDR.COM finds any bad blocks, it will not mount the volume. Instead, it asks you to place another scratch volume in the drive.

You now have multiprocessing console diskettes for the primary and attached processors. Be sure to label them correctly: ATTACHED for the attached processor's console diskette; PRIMARY for the primary processor's console diskette. They are not interchangeable.

You should also indicate on the label the machine for which the multiprocessing console diskettes are intended. These diskettes can only be used in a VAX-11/782 system whose memory configuration is identical to the memory configuration you described when you created the diskettes using BOOTBLDR.COM. These diskettes cannot be used in a single-processor VAX-11/780 system or in a VAX-11/782 system with a different memory configuration.

3.2.2.2 Shutting Down the System

After you have created console diskettes for the primary and attached processors, shut down the system by entering the following command:

```
‡ @SYS#MANAGER:SHUTDOWN
```

This command invokes the system shutdown command procedure, which shuts down the system in an orderly fashion.

Ensure that both processors are halted and that the > > > prompt appears on both console terminals.

3.2.2.3 Booting the VAX-11/782 System

With the system shut down and both processors halted, boot the system in the following manner:

- 1 Insert the primary processor's console diskette in the primary processor's console diskette drive.
- 2 Insert the attached processor's console diskette in the attached processor's console diskette drive.
- 3 Enter the BOOT command on the primary processor's console terminal.
- 4 Log in using the SYSTEM account.
- 5 Enter the DCL command START/CPU on the primary processor's console terminal.
- 6 Enter the BOOT command on the attached processor's console terminal.

The VAX/VMS operating system is now running on the VAX-11/782 system. You should now follow the procedure for editing SYSTARTUP.COM. This procedure is described in Section 3.2.2.4.

3.2.2.4 Editing SYSTARTUP.COM

When the VAX/VMS operating system is running on the VAX-11/782 system, you should edit the site-specific start-up command procedure SYS\$MANAGER:SYSTARTUP.COM to allow automatic restart of the attached processor following a system shutdown.

To accomplish this, edit the command procedure SYS\$MANAGER:SYSTARTUP.COM to include the following commands:

```
‡ START/CPU
‡ WRITE SYS$OUTPUT "You can boot the attached processor now."
```

On a cold start, you can boot the attached processor when the message "You can boot the attached processor now" appears on the primary processor's console terminal. To boot the attached processor, you can issue the BOOT command at the attached processor's console terminal, or you can press the BOOT button on the attached processor's console panel.

3.2.3 Installation Information—VMB Problem and Solution for RX01/TU58

Some changes have been made to an image that resides on the system console known as VMB.EXE. Normally it would be sufficient to simply copy the new image on the console. However, the size of the new image is greatly increased for VMS version 4.4. This means that for some systems that use RX01 or TU58, there may not be enough contiguous space on the console for the new image. A command procedure exists to create the needed space and copy the new image onto the console.

To determine if you need to update your console perform the following:

- 1. Log into the system manager's account.
- 2. Enter the following commands:

```
‡ RUN SYS$SYSTEM:SYSGEN
SYSGEN>CONNECT CONSOLE
SYSGEN>EXIT
‡ EXCHANGE DIR/SIZE CSA1:VMB.EXE
```

- 3. If the size is at least 55 blocks, your console has the correct version of VMB.EXE. If the size is less than 55 block you need to copy the correct version onto the console.

In order to update the console with the new image invoke the console update command procedure UPDATE_CONSOLE.COM as follows:

```
‡ @sys$update:UPDATE_CONSOLE.COM
```

If your system is an 8650, 8600, or 8200 this procedure will simply copy the new file onto your existing console.

If your system is an 11/780, 11/782, 11/785 or 11/750 this procedure will use EXCHANGE to save the contents of your existing console. It will then merge in the new files on the saved copy of your console. Finally it will request that you insert a scratch medium so that it can create a new console containing the new file. Your original console will not be modified.

Problems, Restrictions, and Notes

3.2.4 **VAX/VMS Verify Utility Reference Manual—Text Correction**

On page VER-7, the example should read /READ_CHECK, not /[NO]READ_CHECK. This correction will be incorporated in the next revision of the manual.

3.2.5 **VAX/VMS Developer's Guide to VMSINSTAL—Text Correction**

The VMSINSTAL CHECK_NET_UTILIZATION callback documented in Section 5.2 of the *VAX/VMS Developer's Guide to VMSINSTAL* (a new optional manual) is described as follows:

“ This callback determines whether the net number of free blocks on the VMI\$ROOT device is sufficient to successfully complete the installation. ”

The description should state “peak number” rather than “net number” of free blocks. This correction will be incorporated into the manual in a future revision.

3.2.6 **VAX/VMS Install Utility Reference Manual—Additions and Corrections**

This section describes information not included in the Install Utility documentation.

3.2.6.1 **Enhanced LIST/GLOBAL/FULL Command**

The LIST/GLOBAL/FULL command of the Install Utility now displays the following additional information on global sections:

- Owner and protection
- Access control entries (ACEs) if an access control list (ACL) exists

3.2.6.2 **/SUMMARY Qualifier**

Used with the INSTALL/GLOBAL command, the /SUMMARY qualifier displays a summary of global section and global page usage on the system for both local and shared memory global sections.

3.2.6.3 **Corrections to Text**

Make the following corrections to the *VAX/VMS Install Utility Reference Manual*. These corrections will be incorporated into the next revision of the manual.

- On page INS-1, the format for invoking INSTALL is given as:

```
RUN SYS$SYSTEM:INSTALL
```

This command line format became obsolete with Version 4.0 when the foreign command format was implemented. To establish the INSTALL command as the default for your site, you must define the global symbol INSTALL in your SYLOGIN.COM file as follows:

```
‡ INSTALL = "‡INSTALL/COMMAND_MODE"
```

Once this symbol is defined, you can invoke the Install Utility by entering INSTALL as a DCL command.

In a future release, this format will become the default.

- Page INS-2: Footnote 2 under Example INS-2 should read “with the /OPEN qualifier”, not “with the /SHARED qualifier”.
- Page INS-6: the privilege listed as SYSLCKL should read SYSLCK.
- Page INS-7: the file name GRPCOMMEXE should read GRPCOMM.EXE.
- Page INS-15: in the third paragraph, 00038E should read 0003E8.

3.2.7 ***VAX/VMS Accounting Utility Reference Manual—Corrections***

Make the following corrections to the *VAX/VMS Accounting Utility Reference Manual*. These corrections will be incorporated in the next revision of the manual.

- Page ACC-4: in the example, the keyword ELAPSES should read ELAPSED.
- Page ACC-49: Figure ACC-7 is incorrect. There should be an empty, unused byte at offset 25. ACR\$W_USERNAME should be at offset 26. Each item in the figure should be moved forward by 1 byte, starting with the USERNAME field.

3.2.8 ***Mount Utility Reference Manual—Addition to Manual***

The documentation for the jobwide MOUNT support was omitted from the VAX/VMS documentation. It should read as follows:

In VAX/VMS, any subprocess in the process tree can mount or dismount a volume for the job. When a subprocess mounts a volume (for the job) as a private volume, the master process of the job becomes the owner of this device. This provision is necessary because the subprocess may be deleted and the volume should remain privately mounted for this job.

3.2.9 ***VAX/VMS DECnet Test Sender/DECnet Test Receiver Utility Reference Manual—Text Changes***

The description of the /[NO]DISPLAY qualifier on page DTS-8 should be replaced as follows:

/[NO]DISPLAY=number

Instructs DTS to print the specified number of bytes (in hexadecimal) of data and interrupt messages to DTR. The default is /NODISPLAY.

3.2.10 ***VAX/VMS Authorize Utility Reference Manual—Text Corrections***

Make the following corrections to the *VAX/VMS Authorize Utility Reference Manual*. These corrections will be incorporated in the next revision of the manual.

- Page AUTH-2: the summary of AUTHORIZE commands should include the following qualifiers:
 - /ASTLM (for the ADD command)
 - /GENERATE_PASSWORD (for the MODIFY command)

Problems, Restrictions, and Notes

- Page AUTH-11: in Table AUTH-2, the /FLAG=[NO]PWDEXPIRED function should read /FLAG=[NO]PWD_EXPIRED. Please include the underscore (_).
- On page AUTH-13 of the *VAX/VMS Authorize Utility Reference Manual*, the /PWDEXPIRED and /PWDLIFETIME qualifiers should appear as /[NO]PWDEXPIRED and /[NO]PWDLIFETIME, respectively.
- Page AUTH-14: in the description of the /UIC qualifier, the documentation states that the value of the member number must be in the range of 0-1777776. The correct range is 0-177776.
- Pages AUTH-21, AUTH-37, AUTH-42: the documentation states that the rights identifier values must be in the range 32,768 to 268,435. Note that user-defined identifiers must be in the range 65,536 to 268,435,455. Identifier values of less than 65,536 are reserved.
- Tables AUTH-2 and AUTH-4: the recommended values for process resource limits should read as follows:

Limit	Value
ASTLM	24
BIOLM	18
BYTLM	8192
ENQLM	30
PGFLQUOTA	12800
WSDEFAULT	200
WSQUOTA	500
WSEXTENT	1000

3.2.11 *VAX/VMS Authorize Utility Reference Manual*—Error Messages

The Authorize Utility has the following error messages that have not previously been documented.

BADNODFORM, improper node::remoteuser format

Facility: AUTHORIZE, Authorize Utility

Explanation: You specified the format for the remote node and user incorrectly. The correct format consists of a node name, a pair of colons, and the user name of the remote user. A node name may consist of 1-6 alphanumeric characters and must contain at least one alphabetic character. If you use a wildcard character for either the node or user, you must still include the colons.

User Action: Reenter your command with the correct format.

BADUSR, username does not exist

Facility: AUTHORIZE, Authorize Utility

Explanation: The user name you specified does not exist in the system user authorization file (SYSUAF.DAT).

User Action: Correct the user name and reenter your command. You can display the records in the user authorization file by using the AUTHORIZE command SHOW.

Problems, Restrictions, and Notes

CLIWARNMSG, Warning: /CLITABLES field may need to reflect changes to /CLI field

Facility: AUTHORIZE, Authorize Utility

Explanation: If you modify the command language interpreter (CLI) field of a record in the system user authorization file, you may have to modify the CLITABLES field to reflect the change. If you have set the CLI field to DCL or MCR, however, the CLITABLES field defaults to the correct value.

User Action: If you have changed the CLI field to a value other than DCL or MCR, use the AUTHORIZE command MODIFY/CLITABLES to set the CLITABLES field to the corresponding tables. Refer to the description of the LOGIN Procedure in the *VAX/VMS DCL Dictionary* for further information about specifying CLI tables.

CMDTOOLONG, command line exceeds maximum length

Facility: AUTHORIZE, Authorize Utility

Explanation: The length of your command, after any symbols and logical names have been expanded, exceeds the maximum allowable length.

User Action: Reenter a shorter form of the command.

EXTRAPARM, superfluous parameter detected

Facility: AUTHORIZE, Authorize Utility

Explanation: You have specified too many parameters in the command line. The extra parameter is identified in the message.

User Action: Reenter your command without the excess parameter.

GRANTERR, unable to grant identifier 'id-name' to 'user name'

Facility: AUTHORIZE, Authorize Utility

Explanation: The specified identifier cannot be granted to the specified user. This message should be accompanied by a second message showing the specific reason why the identifier could not be granted.

User Action: Correct the condition identified by the second message and reenter your command.

GRANTMSG, identifier 'id-name' granted to 'user name'

Facility: AUTHORIZE, Authorize Utility

Explanation: The specified general identifier has been granted to the specified user. The user has access to all of the rights associated with the identifier.

User Action: None.

HELPERR, error finding or outputting HELP information

Facility: AUTHORIZE, Authorize Utility

Explanation: An error occurred trying to access the AUTHORIZE HELP file.

User Action: Check that the AUTHORIZE HELP file, by default named UAFHLP.HLB, is located in the proper directory and is not protected against read access.

Problems, Restrictions, and Notes

IDOUTRNG, identifier value is not within legal range

Facility: AUTHORIZE, Authorize Utility

Explanation: The value you specified for an identifier is not within the permissible range. A general identifier may have an integer value between 32,768 and 268,435,455. A UIC identifier takes a value in standard UIC format.

User Action: Reenter your command with an identifier value that is within the permissible range.

INVCMD, invalid command

Facility: AUTHORIZE, Authorize Utility

Explanation: The command you have entered is not a valid AUTHORIZE command.

User Action: Refer to the *VAX/VMS Authorize Utility Reference Manual* for a description of the command you are trying to use and then reenter the command correctly.

INVUSERNAME, username syntax error

Facility: AUTHORIZE, Authorize Utility

Explanation: The user name you specified is invalid due to incorrect syntax. If you are adding a new user name to the system user authorization file with the AUTHORIZE command ADD, the new user name may be 1–12 alphanumeric characters, and it may include underscores.

User Action: Correct the user name and reenter your command.

INVUSERSPEC, error in user specification

Facility: AUTHORIZE, Authorize Utility

Explanation: Your command included an incorrect user specification. In a user specification, you may use a numeric UIC format (for example, [007,007]), an alphanumeric format (for example, [COMPOSERS,HAYDN]), or a user name (for example, HAYDN). You can use wildcards to specify multiple users. Refer to the *VAX/VMS Authorize Utility Reference Manual* for specific syntax rules for the command you are using.

User Action: Correct the user specification and reenter your command.

NAFADDERR, unable to add record to NETUAF.DAT

Facility: AUTHORIZE, Authorize Utility

Explanation: The record you specified could not be added to the network user authorization file (NETUAF.DAT). This message should be accompanied by a VAX RMS message that identifies the specific reason for the error. For example, this error occurs if you try to add a record authorizing a remote user to access more than one local account. Each user at a remote node is allowed access to the files of only one user on the local node.

User Action: If possible, correct the condition identified by the RMS message and reenter your command. Otherwise, examine the network user authorization file to determine why the record could not be added. You can display the contents of the file by using the AUTHORIZE command SHOW/PROXY. You can write the contents of NETUAF.DAT to a listing file by using the AUTHORIZE command LIST/PROXY. If you want to delete

Problems, Restrictions, and Notes

a current record from NETUAF.DAT in order to add a new one, use the AUTHORIZE command REMOVE/PROXY.

NAFAEX, NETUAF.DAT already exists

Facility: AUTHORIZE, Authorize Utility

Explanation: A network user authorization file (NETUAF.DAT) already exists for the local node.

User Action: If you want to create a new network user authorization file, either delete or rename the current one (if you have sufficient privilege to do so). Once the current file has been deleted or renamed, reenter the AUTHORIZE command CREATE/PROXY. Note that you must have sufficient privilege to create a new file.

NAFCREERR, unable to create NETUAF.DAT

Facility: AUTHORIZE, Authorize Utility

Explanation: A network user authorization file (NETUAF.DAT) could not be created. This message should be accompanied by a VAX RMS message that identifies the specific reason why the file could not be created. For example, this error occurs if you do not have sufficient privilege to create the file.

User Action: Correct the condition identified by the RMS message and reenter your command.

NAFDNE, NETUAF.DAT does not exist

Facility: AUTHORIZE, Authorize Utility

Explanation: A network user authorization file (NETUAF.DAT) does not exist on the local node.

User Action: If you have sufficient privilege, use the AUTHORIZE command CREATE/PROXY to create a network user authorization file. Then you can add records to the file by using the AUTHORIZE command ADD/PROXY.

NAFDONEMSG, network authorization file modified

Facility: AUTHORIZE, Authorize Utility

Explanation: The network user authorization file (NETUAF.DAT) has been modified to reflect the change directed by your command.

User Action: None.

NAFNOMODS, no modifications made to network authorization file

Facility: AUTHORIZE, Authorize Utility

Explanation: No modifications have been made to the network user authorization file (NETUAF.DAT).

User Action: None.

NAFUAEERR, entry already exists in NETUAF.DAT

Facility: AUTHORIZE, Authorize Utility

Explanation: The record you have tried to add to the network user authorization file is already in the file; it has not been duplicated.

User Action: None.

Problems, Restrictions, and Notes

NAONAF, unable to open NETUAF.DAT

Facility: AUTHORIZE, Authorize Utility

Explanation: The network user authorization file (NETUAF.DAT) could not be opened. This message should be accompanied by a VAX RMS message that identifies the specific reason for the error. Possible reasons are insufficient privilege, file protection violation, or location of the file in the wrong directory.

User Action: If you do not have sufficient privilege to open NETUAF.DAT, there is nothing you can do except to ask a privileged user, such as your system manager, to access the file for you. If you do have sufficient privilege, make sure the file is located in the proper directory and is not protected against read or write access. Then reenter your command.

NETLSTMSG, listing file NETUAF.LIS complete

Facility: AUTHORIZE, Authorize Utility

Explanation: The contents of the network user authorization file (NETUAF.DAT) have been written to the listing file named NETUAF.LIS.

User Action: None.

NOARG, missing argument for option

Facility: AUTHORIZE, Authorize Utility

Explanation: You specified a qualifier without a required argument.

User Action: Reenter your command and include the required argument.

NODTOOBIG, node name too long

Facility: AUTHORIZE, Authorize Utility

Explanation: VAX/VMS node names cannot exceed six characters. A node name may consist of 1–6 alphanumeric characters; at least one character must be alphabetic.

User Action: Check the node name and reenter your command with the correct name.

NOGRPWILD, wild card group numbers not allowed

Facility: AUTHORIZE, Authorize Utility

Explanation: Wildcard characters are not allowed in the UIC group number field for the command you entered.

User Action: Reenter your command with a specific UIC group number instead of a wildcard character.

NOIDNAM, no ID name was specified

Facility: AUTHORIZE, Authorize Utility

Explanation: The command you entered requires that you include an identifier name.

User Action: Check the *VAX/VMS Authorize Utility Reference Manual* for the syntax rules regarding identifier names for the command you want to use. Then reenter the command including an identifier name.

Problems, Restrictions, and Notes

NOTIDFMT, id name parameter does not translate to ID format

Facility: AUTHORIZE, Authorize Utility

Explanation: The identifier name that you specified does not translate to a corresponding value in general identifier format. Identifier name values translate to either general identifier format or UIC format. General identifier names may be 1 through 31 alphanumeric characters and are stored with an integer value in the range of 32,768 to 268,435,455. General identifiers are created by the AUTHORIZE command ADD/IDENTIFIER.

When you use the AUTHORIZE command GRANT/IDENTIFIER, the first identifier you specify must be in general identifier format. In other words, you cannot grant a UIC-format identifier to another UIC-format identifier.

User Action: Determine why the identifier name does not translate. You can display an identifier name and its corresponding value with the AUTHORIZE command SHOW/IDENTIFIER. To change the value of an identifier name, use the AUTHORIZE command MODIFY/IDENTIFIER.

NOTUICFMT, user id parameter does not translate to UIC format

Facility: AUTHORIZE, Authorize Utility

Explanation: The user specification in your command does not translate to a UIC identifier (an identifier in UIC format).

User Action: Determine why the user specification does not translate. You can display user names and their corresponding UIC values by using the AUTHORIZE command SHOW.

NOUSERNAME, missing username

Facility: AUTHORIZE, Authorize Utility

Explanation: The command you are using requires a user name. A user name is the member name from the alphanumeric form of a user's UIC (user identification code).

User Action: Reenter your command including a user name.

NOUSERSPEC, missing user specification

Facility: AUTHORIZE, Authorize Utility

Explanation: The command you are using requires a user specification. A user specification may be a user name (for example, CAESAR), or a user identification code (for example, [100,44]).

User Action: Reenter your command including a user specification.

PREMMSG, record removed from NETUAF.DAT

Facility: AUTHORIZE, Authorize Utility

Explanation: The record specified in the AUTHORIZE command REMOVE/PROXY has been removed from the network user authorization file.

User Action: None.

Problems, Restrictions, and Notes

PWDNCH, password not changed

Facility: AUTHORIZE, Authorize Utility

Explanation: An error occurred using the random password generator to generate an account password.

User Action: None.

PWDNOL, password not on list; try again

Facility: AUTHORIZE, Authorize Utility

Explanation: The password you specified was not one of those listed.

User Action: Select another password and try again.

RDBADDERR, unable to add 'id-name' to RIGHTSLIST.DAT

Facility: AUTHORIZE, Authorize Utility

Explanation: The identifier name you specified could not be added to the rights database file (RIGHTSLIST.DAT). This message should be accompanied by a VAX RMS message that identifies the specific reason for the error. Most likely, the identifier name already exists in the rights database file. Duplicate identifier names are not allowed in the rights database file.

User Action: Correct the condition identified by the RMS message and reenter your command. If you want to change the name of an identifier in the rights database file, use the AUTHORIZE command MODIFY/IDENTIFIER.

RDBADDERRU, unable to add 'id-name' value: '[UIC]' to RIGHTSLIST.DAT

Facility: AUTHORIZE, Authorize Utility

Explanation: The specified identifier name and its corresponding user identification code (UIC) could not be added to the rights database file (RIGHTSLIST.DAT). This message should be accompanied by a VAX RMS message that identifies the specific reason for the error. Most likely, the identifier name already exists in the rights database file. Duplicate identifier names are not allowed in the rights database file.

This error also occurs if you copy a record in the system user authorization file (SYSUAF.DAT) without specifying a new UIC value for the copy. By default, an identifier name and corresponding UIC value for the new record are written to the rights database file (RIGHTSLIST.DAT); if the UIC has not been changed, it will conflict with the UIC of the original record, and a 'duplicate identifier' error results.

User Action: Correct the condition identified by the RMS message and reenter your command. If you want to change the UIC value of an identifier in the rights database file, use the /VALUE qualifier with the AUTHORIZE command MODIFY/IDENTIFIER. If you copy a record in the system user authorization file, and you want an identifier for the new record to be added to the rights database file, use the /UIC qualifier with the AUTHORIZE command COPY.

Problems, Restrictions, and Notes

RDBADDERRV, unable to add 'id-name' value: 'hex code' to RIGHTSLIST.DAT

Facility: AUTHORIZE, Authorize Utility

Explanation: The specified identifier name and its corresponding integer value (expressed as an 8-bit hexadecimal code) could not be added to the rights database file (RIGHTSLIST.DAT). This message should be accompanied by a VAX RMS message that identifies the specific reason for the error. Most likely, the identifier name or value already exists in the rights database file. Duplicate identifier names or values are not allowed in the rights database file.

User Action: Correct the condition identified by the RMS message and reenter your command. If you want to change the value of an identifier in the rights database file, use the /VALUE qualifier with the AUTHORIZE command MODIFY/IDENTIFIER.

RDBADDMSG, identifier 'id-name' value: 'hex code' added to RIGHTSLIST.DAT

Facility: AUTHORIZE, Authorize Utility

Explanation: A general identifier with the specified name and value has been added to the rights database file (RIGHTSLIST.DAT).

User Action: None.

RDBADDMSGU, identifier 'id-name' value: '[UIC]' added to RIGHTSLIST.DAT

Facility: AUTHORIZE, Authorize Utility

Explanation: A UIC identifier with the specified name and value has been added to the rights database file (RIGHTSLIST.DAT).

User Action: None.

RDBCREERR, unable to create RIGHTSLIST.DAT

Facility: AUTHORIZE, Authorize Utility

Explanation: The rights database file, named RIGHTSLIST.DAT, could not be created. This message should be accompanied by a VAX RMS message that identifies the specific reason for the error. For example, you cannot create another rights database file if one already exists, unless you first delete or rename the original file.

User Action: Correct the condition identified by the RMS message and reenter your command. If you want to create a new rights database file, either delete or rename the current one (if you have sufficient privilege to do so). Once the current file has been deleted or renamed, reenter your command.

RDBDONEMSG, rights database modified

Facility: AUTHORIZE, Authorize Utility

Explanation: The rights database file (RIGHTSLIST.DAT) has been modified.

User Action: None.

Problems, Restrictions, and Notes

RDBMDFYERR, unable to modify identifier 'id-name'

Facility: AUTHORIZE, Authorize Utility

Explanation: The specified identifier could not be modified. This message should be accompanied by a VAX RMS message that identifies the specific reason for the error.

User Action: Correct the condition identified by the RMS message and reenter your command.

RDBMDFYERRU, unable to modify identifier '[UIC]'

Facility: AUTHORIZE, Authorize Utility

Explanation: The specified UIC identifier could not be modified. This message should be accompanied by a VAX RMS message that identifies the specific reason for the error.

User Action: Correct the condition identified by the RMS message and reenter your command.

RDBMDFYMSG, identifier 'id-name' modified

Facility: AUTHORIZE, Authorize Utility

Explanation: The record for the specified identifier in the rights database file has been modified according to the AUTHORIZE command MODIFY /IDENTIFIER.

User Action: None.

RDBNOMODS, no modifications made to rights database

Facility: AUTHORIZE, Authorize Utility

Explanation: The rights database file (RIGHTSLIST.DAT) was not modified.

User Action: None.

RDBREMERR, unable to remove 'id-name' from RIGHTSLIST.DAT

Facility: AUTHORIZE, Authorize Utility

Explanation: The specified identifier could not be removed from the rights database file (RIGHTSLIST.DAT). This message should be accompanied by a VAX RMS message that identifies the specific reason for the error.

User Action: Correct the condition identified by the RMS message and reenter your command.

RDBREMMSG, identifier 'id-name' value: 'hex code' removed from
RIGHTSLIST.DAT

Facility: AUTHORIZE, Authorize Utility

Explanation: The general identifier with the specified name and hexadecimal value has been removed from the rights database file (RIGHTSLIST.DAT).

User Action: None.

Problems, Restrictions, and Notes

RDBREMMMSGU, identifier 'id-name' value: '[UIC]' removed from RIGHTSLIST.DAT

Facility: AUTHORIZE, Authorize Utility

Explanation: The UIC identifier with the specified name and user identification code has been removed from the rights database file (RIGHTSLIST.DAT).

User Action: None.

REVOKEEERR, unable to revoke identifier 'id-name' from 'user name'

Facility: AUTHORIZE, Authorize Utility

Explanation: The specified identifier could not be revoked from the specified user.

User Action: Make sure that the user has been granted the identifier you are trying to revoke. Use the AUTHORIZE commands SHOW/IDENTIFIER/FULL or LIST/IDENTIFIER/FULL to display an identifier and the users who hold it.

REVOKEMSG, identifier 'id-name' revoked from 'user name'

Facility: AUTHORIZE, Authorize Utility

Explanation: The specified identifier has been revoked from the specified user. The user no longer has the rights associated with the identifier.

User Action: None.

RLSTMSG, listing file RIGHTSLIST.LIS complete

Facility: AUTHORIZE, Authorize Utility

Explanation: The contents of the rights database file (RIGHTSLIST.DAT) have been written to the listing file named RIGHTSLIST.LIS.

User Action: None.

SHOWERR, unable to complete show command

Facility: AUTHORIZE, Authorize Utility

Explanation: The AUTHORIZE command SHOW could not be completed. This message should be accompanied by a VAX RMS message that identifies the specific reason for the error.

User Action: Correct the condition identified by the RMS message and reenter your command.

SYMSG2, Error code 'hex code' not found

Facility: AUTHORIZE, Authorize Utility

Explanation: The \$GETMSG system service could not find a corresponding message for the specified error code, which probably indicates that the code is incorrect. Since an incorrect error code obviously should not be generated, this message probably indicates an internal software error.

User Action: Submit a Software Performance Report (SPR) that describes the conditions leading to the error.

Problems, Restrictions, and Notes

WLDNOTALWD, wild card user specs not allowed

Facility: AUTHORIZE, Authorize Utility

Explanation: Wildcard characters are not allowed in the user specification for the command you are using.

User Action: Reenter your command without using wildcard characters.

ZZPRACREN, proxies to 'user name' renamed

Facility: AUTHORIZE, Authorize Utility

Explanation: Proxy access records for the specified user have been renamed to the new user name. When a user name in the system user authorization file (SYSUAF.DAT) is renamed, any records in the network authorization file (NETUAF.DAT) for the original user name are automatically renamed to the new user name.

User Action: None.

ZZSYSPWD, system password modified

Facility: AUTHORIZE, Authorize Utility

Explanation: The system password has been changed to the password directed by your command.

User Action: None.

3.2.12 Authorize Utility—Changes and Notes

This section contains information pertaining to the Authorize Utility.

3.2.12.1 /ACCESS Qualifier

The syntax string for the /ACCESS qualifier to the MODIFY command has been enhanced to allow more readable, flexible usage. The following commands produce identical results.

```
UAF> MODIFY SAM /ACCESS=(primary, 2-3, 5, secondary, 8-12)
UAF> MODIFY SAM /ACCESS="Primary: 2-3, 5; Secondary: 8-12"
UAF> MODIFY SAM /ACCESS=(p,2,s,8,p,3,s,9,p,5,s,10-12)
UAF> MODIFY SAM /ACCESS="2-3 SEC 8-12 PRIM 5"
```

3.2.12.2 /DEFPRIVILEGES and /PRIVILEGES Qualifiers

You can specify the keyword [NO]ALL for the /DEFPRIVILEGES and /PRIVILEGES qualifiers to disable/enable all user privileges.

3.2.12.3 Secondary Passwords

Beginning with Version 4.2, users cannot initially give themselves secondary passwords. The initial setting of the secondary password must be done by the system manager using the Authorize Utility. The reason for this change is to protect careless users who leave their terminal sessions unattended.

In earlier versions of VAX/VMS, anyone could essentially render an account useless by simply adding a secondary password that the account's owner did not know. If a user now tries to initiate a secondary password, the system will respond as follows:

```
‡ SET PASSWORD/SECONDARY
%SET-F-PWD2NOTSET, system manager must initially set secondary passwords
```

3.2.12.4 AUTOLOGIN Flag

A flag named AUTOLOGIN is added to the flags field in the user authorization file (SYSUAF). The flag is set by specifying the qualifier /FLAGS=AUTOLOGIN to one of the following Authorize Utility commands: ADD, MODIFY, or COPY. When set, it makes the account available only by using the autologin mechanism. The following forms of access are disabled:

- Login by any terminal, LAT connection, or SET HOST involving presentation of user name and password
- Access by DECnet task using explicit access control

The following forms of access remain permitted:

- Interactive login by the autologin mechanism
- Batch jobs
- Proxy access by DECnet task

3.2.13 ACCOUNTING Utility—Abbreviated Qualifier Values Restriction

The use of abbreviating qualifier values in the Accounting Utility can produce erroneous results from nonabbreviated qualifiers. For example, the following command produces a display of all summary information and LOGFAILs:

```
* ACCOUNTING/SUMMARY=US !(US abbreviation for USER)
```

However, the identical command used with the nonabbreviated qualifier (USER) produces a display of all summary information without the LOGFAILs:

```
* ACCOUNTING/SUMMARY=USER
```

This problem will be corrected in a future release. Until then, do not use abbreviated qualifiers in the Accounting Utility.

3.2.14 Image Activation, Search Lists, and Known Images—Note

One of the steps involved in image activation uses VAX Record Management Service (RMS) to open the specified image file. When the image to be activated is specified as a logical name, the file specification that is the translation of that logical name is accessed. RMS then opens the image by first attempting to locate the image on one of the known file lists. If the image is not known (that is, the lookup operation fails) then RMS has no other choice but to incur the overhead of locating and opening the image file on disk.

If the image specification includes a semicolon (;) or a period (.) to delimit the version number (whether or not an explicit version number is actually specified) the known file lookup by RMS is skipped. In that case, RMS will always incur the overhead of opening the image file on disk.

The precedence of the known file lookup over the normal file system access during image activation is extended when an image is being activated by way of a search list. For each element on the search list that does not include a file version delimiter, RMS executes a known file lookup. This continues until a lookup is successful or until the search list is exhausted. If the search list is exhausted, RMS then evaluates the entire search list from its beginning a second time in an effort to locate and open the image file on disk. Further

Problems, Restrictions, and Notes

information about locating files using search lists can be found in the *Guide to VAX/VMS File Applications*.

Because of this behavior, it is suggested that care be taken when defining a search list that contains specifications for images that are installed. Regardless of the order of the elements of the search list, the first image in that search list that is found to be installed will be the image selected for activation. That will occur even if there are preceding images in the search list that are not installed.

3.2.15 System Generation Utility (SYSGEN)—Notes and Restrictions

This section contains information related to the System Generation Utility (SYSGEN).

3.2.15.1 UDABURSTRATE Parameter

The UDABURSTRATE parameter is dependent upon configuration and workload. Alteration of the default value can have serious side effects. Consult your DIGITAL Field Service representative before changing the default value of this parameter.

3.2.15.2 SYSGEN Confuses CONSOL.SYS on VAX 11/780 and VAX 11/785 Systems

If you specify a "CONNECT OPA1" at SYSGEN on a VAX 11/780 or VAX 11/785 system, the console software running in the 11/03 will be corrupted. Any attempt to access the console floppy results in an I/O timeout and the floppy being unusable.

Furthermore, if you are attempting to reboot the system from a HSC disk controller, VMB displays the message, "CANNOT FIND CI MICROCODE FILE". The message indicates that although DEFBOO.COM and VMB.EXE can be read without problems, the VMB access path to the front end via the CIB board cannot be used. The only strategy is to turn off the power to the 11/03 subsystem that CONSOLE.SYS to be reloaded.

3.2.16 VAX/VMS System Generation Utility Reference Manual—Text Changes

The following notes document errors and omissions in the Version 4.2 manual:

- The SHARE command is incorrectly documented as SHARE/CONNECT.
- On pages SGN-19 and SGN-20, the examples shown for the CONNECT command are incorrect and should be as follows:

```
SYSGEN> CONNECT LPAO /ADAPTER=3/CSR=%0777514 -  
SYSGEN> /DRIVERNAME=LP2DRIVER/VECTOR=%0200
```

```
·  
·  
·
```

```
SYSGEN> CONNECT NET /NOADAPTER/DRIVER=NETDRIVER
```

- On page SGN-58, the final sentence in the description of the ACP_SHARE parameter should be as follows: "This parameter should be set on when ACP_MULTIPLE is on."
- On page SGN-62, the parameters FREEGOAL and FREELIM are listed as dynamic. These parameters are not dynamic.

- On page SGN-66, the description of the LNMHASHTBL parameter should indicate that the values specified for this parameter are always rounded up to the nearest power of 2. The same is true for the LNMPHASHTBL parameter.
- On page SGN-73, the parameters listed as PQL_DJJQUOTA and PQL_MJJQUOTA are misspelled and should be PQL_DJTQUOTA and PQL_MJTQUOTA respectively.
- On pages SGN-77 and SGN-78, the descriptions of the RMS_DFMBC and RMS_DFNBC parameters should be as follows:

RMS_DFMBC (D)

RMS_DFMBC specifies the default disk block size used by RMS in accessing sequential files.

Normally the default value is adequate.

RMS_DFNBC (D)

RMS_DFNBC specifies a default block count for network access to remote, sequential, indexed sequential, and relative files.

The network block count value represents the number of blocks that RMS is prepared to allocate for the I/O buffers used to transmit and receive data. The buffer size used for remote file access, however, is the result of a negotiation between VAX RMS and the remote File Access Listener (FAL). The buffer size chosen is the smaller of the two sizes presented.

Thus, RMS_DFNBC places an upper limit on the network buffer size that will be used. It also places an upper limit on the largest record that may be transferred to or from a remote file. In other words, the largest record that can be transferred must be less than or equal to RMS_DFNBC multiplied by 512 bytes.

Normally the default value is adequate.

- On page SGN-79, the following information should be included in the description of the SCSNODE parameter:

Specify the parameter value as an ASCII string enclosed in quotation marks ("). Note that the string may not include dollar sign (\$) or underscore (_) characters.

- On page SGN-84, the description of the TTY_DIALTYPE parameter should be as follows:

TTY_DIALTYPE

TTY_DIALTYPE provides flag bits for dial-ups. Bit 0 is 1 for United Kingdom dial-ups and 0 for all others. Bit 1 controls the modem protocol used. Bit 2 controls whether modem lines will hang up 30 seconds after seeing CARRIER if a channel is not assigned to the device. The remaining bits are reserved for future use. See the *VAX/VMS I/O User's Reference Manual: Part I* for more information on flag bits.

3.2.17 User-Created Cluster Quorum File Problem

The cluster quorum file QUORUM.DAT is automatically created by the cluster connection manager when the system is booted for the first time with the SYSGEN parameter DISK_QUORUM specified. The connection manager creates the quorum file with the appropriate attributes and supplies the initial template entry.

Problems, Restrictions, and Notes

Therefore, one should not attempt to create a quorum file manually. Doing so may cause the home block of the quorum disk to be corrupted.

3.2.18 Batch/Print Facility—Notes

The following notes pertain to the Batch/Print facility.

3.2.18.1 SET QUEUE/ENTRY Command—Behavior Change

Specifying the /NOHOLD qualifier with the SET QUEUE/ENTRY command no longer releases jobs specified with /AFTER. You can now use the /NOAFTER qualifier to release a job submitted with /after.

3.2.18.2 Tab Expansion Determined at Start of Queue

When the output queue is started, the VAX/VMS Version 4.4 print symbiont determines if tab expansion is required by accessing the current device characteristics. The Version 4.4 print symbiont expands horizontal tabs only when the device is incapable of handling the tab character. On a device controlled by the LCDRIVER or LPDRIVER, the DCL command SET PRINTER/TAB will set the tab characteristic for that device. On a serial line controlled by the terminal driver, the DCL command SET TERMINAL/TAB will set the tab characteristic for that serial device.

The device characteristics for a particular output queue are determined at the START of that output queue. Therefore, DIGITAL recommends setting the device characteristics before starting the output queue. If the characteristics of a device need to be reset after the output queue has been started, DIGITAL recommends stopping the queue, resetting the device characteristics, and then restarting the output queue. Please be sure the output queue has completely stopped before changing the device characteristics.

3.2.18.3 Generation of Blank Pages When Set-up or Reset Sequence is Specified

In Version 4.0 of VAX/VMS, it was possible to create library setup/reset modules that were output to the device during the processing of the current print job. Setup/reset modules could be output before a specific file, before all files, or after the current job is completed. The VAX/VMS Version 4.0 print symbiont incorrectly inserted form feeds after all setup or reset modules regardless of content. In VAX/VMS Version 4.4, only those modules that insert printable text will be followed by a form feed. No form feed will be inserted after a recognized escape sequence, device control sequence, or operating system command.

DIGITAL realizes that certain limitations exist for output devices that require control sequences in the ASCII range of printable characters. Certain limitations may also exist for those devices that allow the user to reposition output to the top of the page after insertion of printable text. DIGITAL believes this area of the symbiont may require additional flexibility beyond that provided in this functional update. DIGITAL is currently investigating mechanisms by which additional flexibility may be provided.

3.2.18.4 Device Reset Sequence and Form Feed Interaction

Blank pages issued between jobs may be due to interactions between form feed and the device reset escape sequence. Certain programmable devices require the form feed to precede the reset sequence. Extra page problems may be resolved on such devices by inserting a form feed before the reset escape sequence in the device control library module.

3.2.19 User Environment Test Package (UETP)—Restriction Removed

The restriction on running UETP with multiple DR780s has been removed.

3.2.20 VAX RPG II Version 2.0—Restriction

If you currently have VAX RPG II Version 2.0 installed, you must reinstall it after installing or upgrading your system to VAX/VMS Version 4.4.

3.2.21 UDA50—Restrictions on Booting From Multiple UDA50 Systems

If you plan to bootstrap from a UDA50-supported device, there are two restrictions you must keep in mind when you configure your system:

- 1 Each UNIBUS up to (but not including) the one that supports the bootstrap device must have exactly one UDA50. Each UNIBUS, from the bootstrap device upwards, can have up to the legally allowable number of UDA50s.
- 2 You can bootstrap only from the first UDA50 on a UNIBUS (that is, the one with the fixed CSR and vector).

If your system is a VAX-11/750, then the maximum unit number that can be bootstrapped is hexadecimal 'F' (decimal 15).

3.2.22 VAX/VMS System Manager's Reference Manual—VAX-11/730 Dual-RL02 Systems

The VAX-11/730 Dual-RL02 systems are no longer supported as of Version 4.4. Information on customizing the Dual-RL02 system will remain in the *VAX/VMS System Manager's Reference Manual* until the next major software release.

3.2.23 CI Port—Microcode Revision Level Required

The minimum acceptable CI port microcode revision level for either the CI780 or the CI750 is revision level 5.

3.2.24 Time Limit during System Bootstrapping—Restriction

When using SYSBOOT to change system parameters on a system booted from a HSC-controlled disk, there is a limit of four minutes on the time that can be spent at the SYSBOOT> prompt. This limit results from a timeout protocol used between the VAX and the HSC. Please be aware of this time limit and plan your use of SYSBOOT accordingly.

3.2.25 MSCP Server Timeout for UDA Disks—Problem and Workaround

When using the MSCP server to make UDA-controlled disks available to the VAXcluster, add the qualifier `/TIME_OUT=100` to the `SYSGEN MSCP` command that loads the MSCP server.

This qualifier increases the MSCP value controller timeout used by the MSCP server in its dealings with other VAXcluster members. UDA-controlled disks require this larger timeout interval because of differences in the processing of mount requests for drives that are not already spinning.

Failure to increase the timeout interval can result in MSCP Server Detected Error bugchecks on the system that is serving the UDA-controlled disk.

3.2.26 Operational Considerations in VAXclusters—Problems and Solutions

When booting a new system into the cluster, you may see the following warning on the OPA0 terminals at all VAX/VMS systems in the cluster:

```
%PAA0 - Remote system Conflicts with Known System - REMOTE PORT x
```

This message indicates that the newly booted system has either the same SCS node name or SCS system ID (or both the same) as another system already in the cluster. The newly booted system cannot be admitted to the cluster.

It is rather easy to cause this configuration error in either of the following ways:

- Boot a new system with a copy of the system disk of another system in the cluster, and fail to stop in `SYSBOOT` via a conversational boot and set the `SYSGEN` parameters, `SCSNODE` and `SCSSYSTEMID`, to unique values.
- Move the console floppy or TU58 containing a default boot command procedure to boot from a common system directory from a system in the cluster to a new system. Do a default boot of the new system (which normally is set up to not do a conversational boot.) Doing this results in booting the new system with the same `SCSNODE` and `SCSSYSTEMID` as a system already in the cluster.

If this situation should occur, shut down the newly booted system as soon as possible. Then reboot it with properly set `SCSNODE` and `SCSSYSTEMID` parameters. If you leave such a system running and attempting to join the cluster, and if then any sort of virtual circuit failure should occur (for example, a system in the cluster shuts down and reboots), then not all systems will choose to open virtual circuits to the same system of the pair with conflicting name and/or ID.

3.2.27 Tailored Systems and Layered Products—Installation Information

Sites utilizing supported small disk tailored systems must perform an editing operation prior to the installation of any of the VAX/VMS optional software layered products that are listed in this release note. The layered products not in the list may be installed normally.

The layered products involved and the editing operation needed are described in the following paragraphs.

Editing Operation

Before installing any of the optional software layered products in the preceding list, perform the following operation:

- 1. Log into the SYSTEM account
- 2. \$ Set default SYS\$UPDATE
- 3. Using a text editor, edit the file LIBRARY.TLR and remove the line [SYSEXE]SYS.STB
- 4. Exit from the text editor (creating a new version of the file).
- 5. Using a text editor, edit the file REQUIRED.TLR and add the line [SYSEXE]SYS.STB
- 6. Exit from the text editor (creating a new version of the file).
- 7. Log out of the account.

This operation ensures that products requiring the system global symbol table at link time during installation will find the file in the REQUIRED file group. All files that are not members of the REQUIRED file group are tailored to the library disk by VMSINSTAL during installation. The system disk is restored to its original configuration upon completion of the VMSINSTAL command procedure.

Software Products Requiring the Editing Operation

The following list contains the names of optional software products that require the editing operation previously described.

VS5XX DMA DRIVER
DMB32 SYNCHRONOUS DEVICE DRIVER
MICROVMS TSV05 DEVICE DRIVER
IEX-VMS-DRIVER
DRX11-C VMS DRIVER
DRB32 VMS DRIVER AND UTILITIES
VAX DR11-C DRIVER
VAX DRIVER FOR 11C03
VAX-11 TSU05 DEVICE DRIVER
VAX PRODUCER INTERPRETER
VAX-11 RSX
VAX COMMON DATA DICTIONARY
VAXELN ADA
VAX ACMS PRODUCT SET
DECNET/SNA VMS DHCF
VS11-VAX DRIVER
VAX NTR
VAX DRE11-C DEVICE DRIVER
VAX RDB/VMS
ALL-IN-1
VAX 2780/3780 PROTOCOL EMULATOR
VAX DEC/CMS
MUX200/VAX
VAX PRODUCER
VAX SPM
VAX DEC/SHELL

Problems, Restrictions, and Notes

Refer to the "System Software Order Table", SPD Number 28.99.XX for the processor configurations supported by these layered products and their versions.

3.2.28 **TECO—Requires Compatibility Mode**

TECO is a text editor file that is bundled with VAX/VMS. On VAX/VMS systems that do not have built-in compatibility mode hardware (such as VAX 8200, 8300, 8500, and 8800), VAX-11 RSX **must** be installed in order for TECO to be used successfully.

3.2.29 **VAX LISP Version 1.2—Incompatibility with VAX/VMS Version 4.4**

When a VAX LISP user resumes a suspended system, it must be mapped into memory at an exact address. This address is *n* pages beyond where the VAX/VMS RTL ends. In VAX/VMS Version 4.4, the Run-Time Library (RTL) grows beyond the point where a VAX LISP Version 1.2 suspended system needs to start. VAX LISP exits with the fatal error, "VAX LISP image and suspended do not match".

This problem will be corrected in VAX LISP Version 2.0.

Customers who need to use VAX LISP Version 1.2 in the VAX/VMS Version 4.4 environment should contact Margaret Meehan at (617) 568-6515 for more information.

3.2.30 **VAX BASEVIEW—BYTLM Quota**

Before installing VAX BASEVIEW Version 1.0 on VMS Version 4.4, it is necessary to raise the BYTLM quota in the User Authorization File (SYSUAF.DAT).

The steps to do this are:

1. \$ SET DEFAULT SYS\$SYSTEM
2. \$ RUN AUTHORIZE
3. UAF> SHOW SYSTEM

If the SYSTEM account has a BYTLM value of less than 18100, then please raise the value to at least this number. DIGITAL recommends that this be raised slightly higher.

4. UAF> MODIFY SYSTEM/BYTLM=18200
5. UAF> EXIT

You must log out and log back into the SYSTEM account prior to performing the installation, or the raised value will not be in effect, and the installation will fail.

3.2.30.1 **VAX/VMS INSTALL Option N—Compatibility Problem**

Use of the new VMSINSTALL Option N to display or print layered product release notes is not compatible with earlier saved auto answer files created via the VMSINSTALL Option A.

As Option N did not exist on previous versions of VAX/VMS, there was no way that it could be stored in the auto answer file.

As a result, use of an existing auto answer file with Option N will produce the following:

```
%VMSINSTALL-F-AUTOSYNC, Auto-answer file is not in synch with
questions.
-VMSINSTALL-F-AUTOSYNC, question: * Select option [3]:
-VMSINSTALL-F-AUTOSYNC, file: * Do you want to purge files replaced
by this
installation [YES]?
%VMSINSTALL-F-UNEXPECTED, Installation terminated due to
unexpected event.
VMSINSTALL procedure done at 14:00
```

The solution is either not to use an auto answer file with Option N or to recreate the answer file and use Option N concurrently by specifying both options appended together.

The N option allows the installer to view or print, or both view and print the online release notes for those optional layered software products that support online release notes. Note: Currently, not all layered products support online release notes. Use of Option N in these cases will produce no difference in the flow of the installation procedure.

3.2.31 **Guide to VAX/VMS Performance Management—Additions to Manual**

For Version 4.4, the *Guide to VAX/VMS Performance Management* includes the following new materials:

- A description of operations that the system manager may perform after installation to improve overall system performance.
- A new chapter, *Managing System Resources*. This chapter describes procedures for evaluating the performance of the CPU, memory, and disk I/O resources using the Monitor Utility and (to a lesser extent) other standard VAX/VMS utilities. Discussions focus on the utilization of each hardware resource by major VAX/VMS software components and on the measurement, analysis, and possible reallocation of the hardware resources. Suggestions for corrective actions are provided, in case evaluation indicates that improvements are possible.

The chapter includes a tabular summary of important MONITOR statistical items along with rules of thumb for using MONITOR data to evaluate performance of the CPU, memory, and disk I/O resources.

Problems, Restrictions, and Notes

3.2.32 Monitor Utility—Fails in Certain Cases

The Monitor utility fails when monitoring more than one multiprocessor using the following classes: (CLUSTER, SYSTEM, or MODES).

The Monitor utility also fails when PROCESSES/TOPxxx and the SYSTEM class are monitored at the same time.

Both of these problems will be fixed in the next release of VAX/VMS.

3.2.33 Guide to VAX/VMS System Security—Text Changes

Please make the following corrections to the *Guide to VAX/VMS System Security*. These corrections will be included in the next revision of the manual.

3.2.33.1 Defining Ownership Privileges

Section 4.4.2 defines the conditions needed to convey ownership privileges to a user. The numbered list should be replaced with the following:

- 1 Hold the resource attribute to the identifier that owns the file
- 2 Running with BYPASS or SYSPRV
- 3 Running with GRPPRV and in the same group as the file owner

3.2.33.2 Establishing and Changing File Ownership

Section 4.4.5 describes the steps VAX/VMS uses to determine the default owner of a file. These steps should be replaced with the following list:

- 1 An attempt is made to propagate the ownership from a previous version of the file. This will only succeed if the user is privileged (holds BYPASS, SYSPRV, or GRPPRV privilege) or has ownership rights to the owner of the previous version.
- 2 If the attempt to propagate from the previous version fails (either because there is no previous version, the creator lacks ownership rights to the previous version, or the creator is not privileged), then an attempt is made to propagate ownership from the parent directory. This will only succeed if the user is privileged or has ownership rights to the owner of the parent of the directory.
- 3 If the attempt to propagate from the parent directory fails, then the owner of the created file will be the same as the creator of the file.

3.2.33.3 Default ACL Protection

The second sentence in Section 4.5.2.2 states the following:

In addition, when you create a file whose owner identifier is not your UIC, an ACE is added to your ACL for the file that grants full access to your UIC.

This sentence should be replaced with the following corrected version:

In addition, when you create a file whose owner identifier is not your UIC, an ACE is added to your ACL for the file that grants CONTROL access plus the access available to the owner of the file (the Owner field of the SOGW protection mask.)

A similar change will also be made to Section 5.2.6.2 and the flowcharts in Figures 4-4 and 5-5. These changes will be incorporated in the next revision of the manual.

3.2.33.4 Example Change

In Figure 5-10, the line `$ read /end...` should be placed following the line `$ delete /symbol /local /all`. This correction will be included in the next revision of the manual.

3.2.34 TU81-Plus—Caching Capability Note

VAX/VMS will not recognize the caching capability of a TU81-plus tape drive following a reboot of the system. The caching capability will be recognized when the first MOUNT operation following the reboot is performed on the tape drive. Caching will be enabled when the first or any subsequent MOUNT command is issued provided the `/CACHE=TAPE_DATA` qualifier is specified on the MOUNT command.

3.2.35 AIE Ethernet Controller—Unsupported by UETP

The AIE is an Ethernet controller for the BI. There is presently no UETP support for the AIE in VAX/VMS Version 4.4. DIGITAL will provide UETP support in a future release.

3.2.36 ACMS—Restriction

VAX ACMS Version 1.2 will cause your VAX/VMS system to be unusable if you try to run on VAX/VMS Version 4.4. Because of a problem latent in Version 1.2, ACMS fails to release locks on SYSUAF.DAT after checking user authorization. Until now, RMS has been silently handling the problem at image rundown. With VAX/VMS Version 4.4, however, RMS will leave the ACMS locks in place, so that once a user has logged into ACMS, no subsequent users will be able to log into VAX/VMS.

This problem has been fixed in ACMS Version 2.0, and ACMS Version 2.0 kits will be available before VAX/VMS Version 4.4.

To avoid difficulty, follow these guidelines:

- Install ACMS Version 2.0 **before** you install VAX/VMS Version 4.4.
- If you do not plan to upgrade to ACMS Version 2.0, **do not** install VAX/VMS Version 4.4.

3.2.37 Microfiche Listings—Applies to Customers Currently Under Service Contract

The microfiche listings for VAX/VMS Version 4.4 supersede those distributed with Versions 4.0 through V4.3 to reflect the fact that all facilities of VMS except the SYS facility have been rebuilt for Version 4.4. For the SYS facility, the microfiche contains listings of the Version 4.0 executive plus listings of the patch files that were used to patch SYS.EXE in Versions 4.0 through 4.4. These patch files are named SYSV4*.VUP in the SYS facility. Finally, for proprietary reasons, certain facilities and listings of selected modules in other facilities have been removed from the microfiche kit.

Problems, Restrictions, and Notes

3.2.38 TMPJNL and PRMJNL—Removed

The TMPJNL and PRMJNL privileges, which were never used by VAX/VMS, have been removed from VAX/VMS Version 4.4.

Any documentation which may still mention these privileges will be updated to reflect this change in the next release.

3.3 Application Programmer Information

This section contains information about the VAX/VMS Version 4.4 release that pertains to application programmers.

3.3.1 *Guide to Programming on VAX/VMS*—Change in Focus

The focus of the *Guide to Programming on VAX/VMS* has changed to supply programmers in **any** programming language with guidelines for using the development tools available with VAX/VMS. Therefore, “FORTRAN” in the title has been eliminated and the emphasis on FORTRAN has been lessened. However, the manual still has a FORTRAN orientation, all examples are in FORTRAN, and the programmer is assumed to have a reading knowledge of the FORTRAN language.

3.3.2 *VAX/VMS Linker Reference Manual*—Correction

Please note the following corrections to the *VAX/VMS Linker Reference Manual*:

- On page LINK-1, the default for the command qualifier `/[NO]USERLIBRARY` should read `/USERLIBRARY=ALL`. This correction will be incorporated in the next revision of the manual.
- The reference to Section 6.3.6.2 on page LINK-31 (third list item at the top of the page) is incorrect. The correct reference is to Section 5.3.6.2.
- The reference to Appendix A on page LINK-61 (fourth line, second paragraph from the bottom) is incorrect. The correct reference is to Chapter 6, which contains information on the VAX Object Language.
- Example 3 on page LINK-129 is incorrect. The example should read as follows:

```
‡ LINK LAMAR,SYS$INPUT/OPTION
  GRABLE/SHAREABLE
```

Note, GRABLE is the name of a shareable image file and not an options file as previously documented. The above correction also applies to Example 2 on page LINK-142.

3.3.3 *Run-Time Library Routines Reference Manual*—Additions

Documentation has been added to Chapter 3 of the *VAX/VMS Run-Time Library Routines Reference Manual* concerning user-written exit handlers for screen management routines. This documentation explains why pasteboards and virtual keyboards cannot be deleted from within a user-written exit handler.

3.3.4 Run-Time Library Screen Management Facility—Restriction

Due to changes made to the Screen Management Facility, the following restriction now applies to the routines `SMG$SET_BROADCAST_TRAPPING`, `SMG$ENABLE_UNSOLICITED_INPUT`, and `SMG$SET_OUT_OF_BAND_ASTS`. For AST routines written in a language that does not support optional parameters (for example VAX BASIC), all system parameters must be specified. This restriction is illustrated in the example for the `SMG$DISABLE_BROADCAST_TRAPPING` routine in the *VAX/VMS Run-Time Library Routines Reference Manual*.

3.3.5 Debugger—Problems and Restrictions

Note the following restrictions and problems that apply to the Debugger Utility.

3.3.5.1 Debugging Shareable Images—Restriction

Support for debugging shareable images is new with Version 4.4 and is described in Chapter 4 of the *VAX/VMS Debugger Reference Manual*.

There is one restriction you should be aware of when debugging a shareable image: it must have been linked with the `/DEBUG` qualifier. If the image was not linked with the `/DEBUG` qualifier, you will still be able to “SET IMAGE” to that image, but then you may obtain incorrect results.

The behavior is summarized in the following table for an arbitrary shareable image, X.

Command	Effect
<code>LINK/SHARE/DEBUG X</code>	You can SET IMAGE to X and debug it as documented.
<code>LINK/SHARE X</code>	You can SET IMAGE to X, but you may obtain incorrect results when you try to debug it (when using SET BREAK, EXAMINE, and so on). This problem will be corrected in a future release of the debugger.
<code>LINK/SHARE/NOTRACE</code>	You cannot SET IMAGE to X and, therefore, cannot debug it with debugger commands.

3.3.5.2 Debugging SMG Programs—Problem

The debugger now uses the VAX/VMS Screen Management Facility (SMG) to implement screen mode. If your program also calls SMG routines, and you debug it with the debugger running on the same terminal, there will probably be interference between your program and the debugger.

To avoid this problem, debug the program using two terminals. The technique is described in Appendix D of the *VAX/VMS Debugger Reference Manual*.

Problems, Restrictions, and Notes

3.3.5.3 Debugger Changes Affecting Compatibility with Earlier Versions

This section notes any changes in the debugger for Version 4.4 that are incompatible with Version 4.2 or earlier versions. All changes concern the display of various windows in screen mode.

New Display Window Definitions

Changes to the built-in window definitions and the addition of a PROMPT predefined display have caused some incompatibilities with earlier versions of the debugger. If you use built-in window definitions, such as H2, in your debugger initialization file or in your own command or key definitions, then you should be aware of the following changes:

- Previously, the bottom sixth of the screen (lines 21-24 on a VT100 or VT200 series terminal) could not be used for defining windows. That area was reserved for the debugger prompt, debugger input, debugger diagnostic messages, and program output. Now, windows can occupy any part of the screen, and the new PROMPT display shows the debugger prompt, debugger input, and program output.
- The boundaries of the built-in windows have been redefined to cover the greater usable screen height. For example, on a VT100 or VT200 series terminal, FS (full screen) now covers lines 1-24, H1 lines 1-12, H2 lines 13-24, and so on. And a new symbol prefix, S, denotes a multiple of one sixth of the screen.
- The PROMPT display occupies window S6 by default (bottom sixth of screen). Note that, to avoid confusion, the PROMPT display is always on top of the display “pasteboard” and, therefore, will hide the part of any display that overlaps the PROMPT window.
- By default, the OUT display is now at S45 (not, as previously, at H2), so it will not be hidden by the PROMPT display. And the keypad keys that manipulate windows have been redefined so that no display is positioned behind S6.

If your debugger initialization file contains DISPLAY or SET DISPLAY commands to locate displays near the bottom of the screen (for example, at H2, T3, or Q34) you may want to modify these window definitions so the displays will not be hidden.

New Keypad Key Definitions

The debugger’s built-in definitions for keypad keys KP7 and MINUS have been changed to accommodate changes in the built-in window definitions. The new key definitions are as follows:

```
KP7          = DISPLAY SRC AT LH1, INST AT RH1, OUT AT S45, PROMPT AT S6
GOLD KP7     = DISPLAY INST AT LH1, REG AT RH1, OUT AT S45, PROMPT AT S6
BLUE KP7     = Not defined
MINUS        = DISPLAY %NEXTDISP AT S12345
GOLD MINUS   = Not defined
BLUE MINUS   = DISPLAY SRC AT H1, OUT AT S45, PROMPT AT S6 (this is the
               default for high-level languages)
```

Refer to the previous section for information about the new window definitions.

Register Display

The predefined register display (REG) has been reformatted to take advantage of the new window capabilities. REG is now a square display that fits in one of the quarters of the screen (for example, the top left-hand window LH1 or the top right-hand window RH1). If your debugger initialization file had a command like "DISPLAY REG AT Q3", then you may want to change it to something like "DISPLAY REG AT RH1" to accommodate the reshaped register display.

Display Kind "NORMAL" Renamed to "OUTPUT"

The display kind "NORMAL" has been renamed "OUTPUT". If your debugger initialization file contains DISPLAY or SET DISPLAY commands that specify a display kind, you may want to change any occurrences of "NORMAL" to "OUTPUT". However, the display-kind name "NORMAL" will continue to be accepted as a synonym for "OUTPUT".

3.3.6 Terminal Driver Support—Problems

3.3.6.1 SET HOST/DTE/DIAL Command—DMF-32 Problem and Workaround

SET HOST/DTE/DIAL does not work with the DMF-32 controller. The problem is that the modem sends a response character to the host when it detects a carrier signal, but the DMF-32 drops any input until it sees the carrier signal.

One workaround is to modify the example autodialer provided in SYS\$EXAMPLES:DT_DF03.MAR to perform a IO\$_SENSEMODE!IO\$_M_RD_MODEM \$QIO to check for a carrier signal. If set, the autodialer should assume success and continue.

3.3.6.2 SET HOST/DTE/LOG Command—Log File Problem

There is a known problem with the SET HOST/DTE/LOG command. The log file that is created may include extra line-feed characters. This problem will be corrected in a future update to VAX/VMS.

3.3.7 VAX/VMS Command Definition Utility Reference Manual—Example Correction

The following example is an excerpt from Example CDU-2 in the VAX/VMS Command Definition Utility Reference Manual.

To make this BASIC program execute as described in the documentation, change the following lines (comments describe the changes):

```

200 SUB EXIT_COMMAND !Same as documented.
    !exclude EXTERNAL INTEGER FUNCTION SYS$EXIT
    CALL SYS$EXIT(1% BY VALUE) !Note addition.
290 SUBEND !Same as documented.
1  EXTERNAL INTEGER FUNCTION CLI$DCL_PARSE,CLI$DISPATCH !Exclude LIB$GET_INPUT
    EXTERNAL INTEGER FUNCTION SEND_COMMAND,SEARCH_COMMAND,EXIT_COMMAND !Same
    EXTERNAL INTEGER TEST_TABLE,LIB$GET_INPUT !Note addition.
2  IF NOT CLI$DCL_PARSE(,TEST_TABLE,LIB$GET_INPUT,LIB$GET_INPUT,'TEST> ')
    AND 1%
    THEN GOTO 2 !Note elimination of 0 above.

```

3.3.8 *VAX Text Processing Utility Reference Manual*—Documentation Changes

The following are corrections to the documentation for VAXTPU.

3.3.8.1 **GET_INFO—Restriction**

The material in the *VAX Text Processing Utility Reference Manual* does not include a restriction on using the built-in procedure GET_INFO. The following material should be added to the manual's description of the built-in procedure GET_INFO:

Be careful when you write programs that attempt to search one of the lists maintained by VAXTPU. VAXTPU provides only one context for traversing each list. VAXTPU maintains lists of buffers, defined keys, key maps, key-map lists, processes, and windows. You can search a list by using "first", "next", "previous", "current", or "last" as the second parameter to the built-in procedure GET_INFO.

If you create nested loops that attempt to search the same list, the results are unpredictable. For example, a program attempting to search two key-map lists for common key maps may contain the built-in procedure GET_INFO (KEY_MAP, "next",...) in a loop within a loop containing GET_INFO (KEY_MAP, "previous",...). This creates an infinite loop.

3.3.8.2 **VAX BLISS—VAXTPU Example**

The VAX BLISS example TPU-1 in Section 12 of the *VAX/VMS Utility Routines Reference Manual* contains errors. The following example contains corrections to Example TPU-3.

Example 3-1 Sample VAX BLISS Template for Callable VAXTPU

```

! How to declare the VAXTPU routines
external routine
    tpu$FILEIO,
    tpu$HANDLER,
    tpu$INITIALIZE,
    tpu$EXECUTE_INIFILE,
    tpu$EXECUTE_COMMAND,
    tpu$CONTROL,
    tpu$CLEANUP;

! How to declare the VAXTPU literals
external literal
!
! File I/O operation codes
    tpu$k_close,
    tpu$k_close_delete,
    tpu$k_open,
    tpu$k_get,
    tpu$k_put,
!
! File access codes
    tpu$k_access,
    tpu$k_io,
    tpu$k_input,
    tpu$k_output,
!
! Item codes
    tpu$k_calluser,
    tpu$k_fileio,
    tpu$k_outputfile,
    tpu$k_sectionfile,
    tpu$k_commandfile,
    tpu$k_filename,
    tpu$k_journalfile,
    tpu$k_options,
!
! Mask for values in options
    tpu$m_recover,
    tpu$m_journal,
    tpu$m_read,
    tpu$m_command,
    tpu$m_create,
    tpu$m_section,
    tpu$m_display,
    tpu$m_output,
!
! Bit positions for values in options
    tpu$v_display,
    tpu$v_recover,
    tpu$v_journal,
    tpu$v_read,
    tpu$v_create,
    tpu$v_command,
    tpu$v_section,
    tpu$v_output,

```

(Continued on next page)

Problems, Restrictions, and Notes

Example 3-1 (Cont.) Sample VAX BLISS Template for Callable VAXTPU

```
!  
! VAXTPU status codes  
    tpu$_nofileaccess,  
    tpu$_openin,  
    tpu$_inviocode,  
    tpu$_failure,  
    tpu$_closein,  
    tpu$_closeout,  
    tpu$_readerr,  
    tpu$_writeerr,  
    tpu$_success;  
  
own  
    OPTIONS:      bitvector [32];  
! OPTIONS will be passed to VAXTPU  
GLOBAL ROUTINE top_level =  
BEGIN  
!++  
! Main entry point of your program  
!--  
! Your_initialization_routine must be declared as a BPV  
local BPV:      vector[2,long] initial (TPU_INIT,0);! Procedure block  
! First establish the condition handler  
LIB$ESTABLISH (tpu$handler);  
  
! Call the initialization routine and pass it the address of the BPV  
! which has the address of your initialization routine (VAXTPU  
! calls this)  
tpu$initialize (BPV);  
! Use the following call if the options word passed to VAXTPU indicated that  
! an initialization file needs to be executed and/or the TPU$INIT_PROCEDURE  
! in the section file needs to be executed.  
tpu$execute_inifile();  
  
! Let VAXTPU take over.  
tpu$control();  
! To break out of VAXTPU, use call_user from within a VAXTPU program  
! Upon return from tpu$control, the editing session is done  
tpu$cleanup();  
! Loop and start the sequence over or exit  
return tpu$_success;  
  
END;  
  
ROUTINE TPU_INIT =  
BEGIN  
!  
!--  
own    BPV:      vector[2,long] initial (TPU_IO,0);! Procedure block  
Macro  
    OUTFILE_NAME= 'OUTPUT.TPU%',  
    COMFILE_NAME= 'TPUINI.TPU%',  
    SECFILE_NAME= 'SYS$LIBRARY:EVESECINI.TPU$SECTION%',  
    FILE_NAME= 'FILE.TPU%';
```

(Continued on next page)

Example 3-1 (Cont.) Sample VAX BLISS Template for Callable VAXTPU

```

! Set VAXTPU options I want to enable
OPTIONS[tpu$v_display] = 1;
OPTIONS[tpu$v_section] = 1;
OPTIONS[tpu$v_create] = 1;
OPTIONS[tpu$v_command] = 1;
OPTIONS[tpu$v_recover] = 0;
OPTIONS[tpu$v_journal] = 0;
OPTIONS[tpu$v_read] = 0;
OPTIONS[tpu$v_output] = 1;
begin
    !Just for BIND
bind
! Set up item list to pass back to VAXTPU to tell it what to do
! VAXTPU calls me back later
ITEMLIST = uplit byte (
!buffer length,      item code,      buffer address,  return address
word (4),  word (tpu$k_options),  long (OPTIONS),  long (0),
word (4),  word (tpu$k_fileio),   long (BPV),      long (0),
word (%charcount(outfile_name)),
                word (tpu$k_outputfile),  long (uplit(%ascii(outfile_name))),
                                                long (0),
word (%charcount(comfile_name)),
                word (tpu$k_commandfile), long (uplit(%ascii(commandfile_name))),
                                                long (0),
word (%charcount(file_name)),
                word (tpu$k_filename),   long (uplit(%ascii(file_name))),
                                                long (0),
word (%charcount(secfile_name)),
                word (tpu$k_sectionfile), long (uplit(%ascii(secfile_name))),
                                                long (0),
long (0) );
return ITEMLIST;
end;
END;
                ! End of routine TPU_INIT

GLOBAL ROUTINE TPU_IO (P_OPCODE, FILE_BLOCK, DATA: ref block [,byte]) =
BEGIN
!
!--
local
    item: ref block [3,long],      ! Item list entry
    status;

! Look at the opcode (operation) that VAXTPU wants me to perform
! and if I don't want to do it, just call it back
! if (.P_OPCODE NEQ tpu$k_open)
!     then
!         return (tpu$fileio (.p_opcode, .file_block, .data));
!
! Else set what operation to do

selectone ..P_OPCODE of
set
[tpu$k_open]:
! Time to open a file
!
    begin
        item = .data;          ! Point to the FILENAME item list entry
        end;
        return tpu$_success;   ! End of tpu$k_open
    end;
[tpu$k_get]:
! If none exists, then no data
! Time to read a record
    begin
        end;

```

(Continued on next page)

Problems, Restrictions, and Notes

Example 3-1 (Cont.) Sample VAX BLISS Template for Callable VAXTPU

```
[tpu$k_put]:                ! Time to write a record
begin
    return tpu$_success;
end;
[tpu$k_close]:             !Time to close a file
begin
    return tpu$_success;
end;
[tpu$k_close_delete]: lib$stop (...p_opcode);
[otherwise]: lib$stop (...p_opcode);
tes;
return tpu$_success;
END;                        ! End of routine TPU_IO
```

3.3.9 Run-Time Library Support of VAX BASIC—USEROPEN Problem

In VAX/VMS Version 4.4, the Run-Time library support for VAX BASIC clears the RAB\$V_WAT and RAB\$V_TMO bits of the RAB\$L__ROP field. This occurs each time a FIND or GET is executed. Applications that use a USEROPEN to set either of these bits and expect them to stay set will not work properly under VAX/VMS Version 4.4. This problem will be fixed in a future release of VAX BASIC.

3.3.10 PL/I PRINT FILE Format—Line-Feed Change

Prior to VAX/VMS Version 4.2, VAX PL/I generated an extra line-feed immediately following a PAGE directive for PRINT format files. This extra line is no longer generated when PL/I programs are run on VAX/VMS versions later than Version 4.2. Applications that require the old behavior can approximate it using a PUT SKIP command when the ENDPAGE condition is raised, or when a PAGE is explicitly output.

While DIGITAL recommends that /NOFEED be used for printing formatted files, this change should allow PL/I PRINT files that are generated on a VAX/VMS Version 4.2 or later system to be printed on forms with the same number of lines per page as those of the print file using /FEED.

Note that the effect of this change may show up in different ways depending upon the printer type. New printers and terminal devices will simply print everything one line higher on the page. Older line printers may ignore some line-feeds at the top of page so that this change will only show up when the first line of text is printed part way down the page.

3.3.11 **VAX Ada—Compatibility Problem With VAX/VMS Debugger**

If users at your site heavily use VAX DEBUG with VAX Ada programs, you may want to consider delaying installation of this version of VMS until you have obtained the next maintenance release of the Ada compiler after version 1.2.

This release of VMS includes some improvements in the debugger and linker that interfere with the debugging of Ada programs compiled with VAX Ada Version 1.2, or earlier. The problem is that arrays and records whose layout is only known at run time cannot be debugged.

3.4 **System Programmer Information**

This section contains information of interest to the system programmer.

3.4.1 **DR32 Microcode Loader—Problem Fixed**

In Version 4.3, a parity error was reported if a logical name of the form XFc\$WCS was used to load microcode for a DR32. This problem has been fixed.

3.4.2 **VAX/VMS I/O User's Reference Manual: Part II—Additional Information about DR32 Microcode Loader**

The following information should be added in Section 4.4.4 of the *VAX/VMS I/O User's Reference Manual: Part II* immediately after item 2 ("If the opening procedure...").

"By default, XFLOADER attempts to load microcode into all DR32s on a system. To limit microcode loading to a subset of DR32s, define the logical name XF\$DEVNAM using the device names of the DR32s as the equivalence names. XFLOADER will search for the translation using the LNM\$FILE_DEV search list. For example:

```
‡ DEFINE/SYSTEM XF$DEVNAM XFA0,XFC0
```

This definition tells XFLOADER to load microcode only in the first and third DR32s on the system."

The sentence that will immediately follow the above addition "After loading microcode into all available DR32s,..." should be changed to read "After loading microcode into all specified DR32s,..."

3.4.3 **VAX MACRO and Instruction Set Reference Manual—Additional Information on Cyclic Redundancy Check (CRC)**

The following step should be included in the Cyclic Redundancy Check (CRC) instruction on page 9–138 of the *VAX MACRO and Instruction Set Reference Manual*.

Problems, Restrictions, and Notes

Upon completion of the CRC instruction,
registers R0, R1, R2 and R3 are left as follows:

R0 = result of CRC
R1 = 0
R2 = 0
R3 = address one byte beyond end of source string

3.4.4 \$PRDEF Symbols—Documentation Addition

The documentation for the \$PRDEF symbols was omitted from previous release notes. It should read as follows:

The following internal processor registers (IPRs) are no longer common to all VAX processors. Their definitions have been removed from \$PRDEF:

- NICR—Interval Clock Next Interval Register
- ICR—Interval Clock Interval Count Register
- TODR—Time of Day Register
- ACCS—Accelerator Control Status Register
- ACCR—Accelerator Reserved
- PME—Performance Monitor Enable

New CPU-specific processor register definition macros have been added to LIB.MLB to define the CPU-specific IPRs. The macro names have the format \$PRxxxDEF, where xxx is the number associated with the processor (for example, \$PR780DEF will define PR780\$_ACCS).

The only legitimate references to these registers are in CPU-dependent code. These references must use the new CPU-dependent IPR definitions.

Note, however, that time-wait loops must **never** directly reference the clocks. They **must** use a time-wait macro that is CPU independent. A new, CPU-independent time-wait macro called TIMEDWAIT has been added to LIB.MLB. This should eliminate any need for hand-coded, time-wait loops.

There should no longer be any references to PR\$_ICR or PR\$_TODR to do time-wait loops. TIMEDWAIT allows for up to six special-purpose instructions to be placed in its timing loop. However, the loop timing is based on having one BITx and one conditional branch instruction embedded within the loop. Therefore, if you have a loop with no embedded instructions, you may want to adjust the TIME argument accordingly. A good rule of thumb is to add 25 percent to the **time** argument if the loop has no embedded instructions.

To reference PR\$_TODR for logging purposes, use EXE\$READ_TODR and EXE\$WRITE_TODR. These two new loadable, CPU-dependent routines have been added for code that must reference this type of value.

3.4.5 DR32 Bitfield Definitions—Corrections

The macro definition \$XFDEF in SYS\$LIBRARY:STARLET.MLB defines values necessary for programs that call the DR32 driver, the "XFDRIVER". These VAX MACRO definitions are duplicated in a file called SYS\$LIBRARY:XFDEF.FOR for those who call the DR32 driver from a FORTRAN program. In VAX/VMS Version 4.4, some of the error bit definitions in the second longword of the IOSB and some of the error bit definitions in the DR32 status longword in the DR32 command packet have been changed in XFDEF.FOR to match those in \$XFDEF.

Specifically, the values for the following fields in the second longword of the IOSB have changed: XF\$M_IOS_CIPE, XF\$M_IOS_DIPE, XF\$M_IOS_PARERR, XF\$M_IOS_RDSERR, and XF\$M_IOS_WCSPE; and the following definitions in the DR32 status longword have changed: XF\$M_PKT_CMDSTD and XF\$M_PKT_DRVABT.

If you have a FORTRAN program that calls the DR32 driver, it is suggested that you recompile your program under VAX/VMS Version 4.4.

3.4.6 LPA11-K Driver (LADRIVER)—Changing Timeouts Allowed

The driver for the LPA11-K (LADRIVER) times out all \$QIOs after two seconds if they have not completed. The driver does not provide any parameters that allow the user to change the length of the timeout.

In VAX/VMS Version 4.4 and later, the timeout period applied to all \$QIOs can be changed with the following patch commands executed from a suitably privileged account:

```

$ PATCH SYS$SYSTEM:LADRIVER.EXE/OUTPUT=SYS$SYSTEM:LADRIVER.EXE
PATCH> SET ECO 25
PATCH> REPLACE/INSTRUCTION LA$TIMEOUT_VALUE
OLD> 'PUSHL      I`#00000002'
OLD>EXIT
NEW> 'PUSHL      I`#0000003C'
NEW>EXIT
PATCH>UPDATE
PATCH>EXIT

```

Substitute the desired timeout value for the "0000003C" in the example above. When you reboot, the system will load the new copy of the driver containing the new timeout value.

3.4.7 CPUDISP Macro—Format Restriction

One format previously supported by the CPUDISP macro prior to VAX/VMS Version 4.4 is no longer allowed. An example taken from the XFDRIVER follows.

Old CPUDISP invocation:

```

CPUDISP <DR_780,DR_750,DR_730,DR_790>
; * Dispatch on CPU type *

```

The new CPUDISP invocation must be in the form of 2-tuples, where a 2-tuple is the CPU designator (for example 780, UV1, etc.) and the macro label that begins the code specific to that CPU (for example DR_780).

Problems, Restrictions, and Notes

New CPUDISP invocation:

```
CPUDISP <-  
  <780,DR_780>,-  
  <750,DR_750>,-  
  <730,DR_730>,-  
  <790,DR_790>-  
> ; * Dispatch on CPU type *
```

Index

A

- Accounting Utility (ACCOUNTING) • 3-17
 - abbreviated qualifier values • 3-29
- ACL Protection • 3-38
- Alternate root
 - creating on a common system disk • 1-29
 - creating on common system disk • 1-30
- ANALYZE/RMS_FILE • 2-14
- Authorize Utility (AUTHORIZE) • 2-6, 3-17
 - AUTOLOGIN flag • 2-7, 3-29
 - /ACCESS qualifier • 2-6, 3-28
 - /PWDEXPIRED qualifier • 3-18
 - /PWDLIFETIME qualifier • 3-18
 - /secondary passwords • 2-6, 3-28
- AUTOGEN.PAR parameter file
 - creation of • 1-20

B

- Bad Block Locator Utility (BAD)
 - to check new console volume • 1-14, 1-16, 1-18
- BASIC error messages • 2-22
- Batch/Print facility • 2-5
- BOOTBLDR.COM • 3-10
- BOOT command • 3-14, 3-15
- Bootstrapping time limit • 3-33

C

- Callable interface
 - for VAXTPU • 2-2
- CDU (Command Definition Utility) • 3-43
- CI port driver (PADRIVER) • 2-25
- Cold start • 3-15
- Command procedure
 - BOOTBLDR.COM • 3-10
 - default bootstrap • 3-12
 - DMOBOO.COM • 3-11
 - SHUTDOWN.COM • 3-14
 - SYSTARTUP.COM • 3-15

Console volume

- building new site-specific • 1-14, 1-16, 1-18
 - build to boot from SYSF • 1-14, 1-16, 1-18
- CRC (Calculate Cyclic Redundancy Check)
instruction
- MACRO • 3-49
-

D

- Debugger
 - new features • 2-13
 - problems and restrictions • 3-41
- DECnet-VAX
 - DTS/DTR • 3-17
 - shutting down for upgrade • 1-9
- Decompression of library
 - space requirement • 1-22
- Default file type
 - for VAXTPU section files • 2-2
- Default key map
 - in EVE interface • 2-2
- DEFINE/FORM command • 2-5
- Device code
 - format • 1-11
- Device control library module • 2-5
- Disk space
 - requirements for upgrade • 1-6
- DMOBOO.COM • 3-11
- Driver • 2-24

E

- EDT Keypad Emulator
 - changing the the default interface • 2-7
- Error Log Utility • 2-14
- EVE\$CLEAR_KEY pre-defined EVE procedure • 2-2
- EVE\$INIT_KEY pre-defined EVE procedure • 2-2
- EVE editor
 - changing the default interface • 2-7
 - key maps in • 2-2
 - pre-defined procedures in • 2-2
- Extensible VAX Editor
 - See EVE

Index

F

- File names • 3-4
- File Ownership • 3-38
- File type
 - for section files
 - change in default • 2-2
- Floppy diskette
 - on VAX-11/782
 - building • 3-6
- Form
 - specifying for a queue • 2-5

G

- GET_INFO built-in procedure
 - string constants • 2-2, 3-44

I

- I/O routine
 - user supplied
 - change to VAXTPU callable interface for • 2-2
- Image Activation • 3-29
- INITIALIZE/QUEUE command • 2-5
- Installation
 - mandatory update • 1-21
- Installation summary
 - system upgrade • 1-4
 - upgrade/optional product • 1-4
- Install Utility (INSTALL) • 3-16
 - for VAXTPU shareable images • 2-7
 - invoking • 3-16
 - LIST/GLOBAL/FULL command • 3-16
 - /GLOBAL/SUMMARY qualifier • 3-16

L

- Line-Feed • 3-48
- Linker • 2-13, 3-40
 - linking shareable image • 2-13
- LIST/GLOBAL/FULL command • 3-16
- LPA 11-K timeouts • 3-51

M

- Machine check • 3-11
- MACRO
 - Cyclic Redundancy Check (CRC) • 3-49
 - documentation change • 3-49
- Mail Utility (MAIL) • 3-3
 - in a mixed cluster • 3-3
 - /SELF qualifier • 3-3
- MAKEROOT
 - how to invoke • 1-29
 - requirements to execute • 1-29
 - use in creating alternate roots • 1-29
- Mandatory update • 1-21
- Memory configuration
 - determination of • 3-7
- Microcode
 - error in • 3-10
- Mount Utility (MOUNT) • 3-17
 - CACHE=TAPE_DATA qualifier • 2-11
 - change in jobwide support • 3-17

N

- Networking • 2-3
- Node name • 2-3

P

- Page file size
 - use in making alternate root • 1-30
- PL/I PRINT FILE • 3-48
- Print Symbiont (PSM) • 2-15
 - device reset sequence • 3-32
 - generation of blank pages • 3-32
 - tab expansion at start of queue • 3-32
- PRINT USING Function • 2-16
- Privileges • 3-38

Q

- Queue
 - during upgrade • 1-7
 - specifying forms • 2-5
 - specifying stock • 2-5

R

- Rebooting
 - from BOOT58 level • 1-15, 1-17
 - to restart upgrade • 1-14, 1-17
- Removing files after upgrade
 - instructions for • 1-22
- Restart
 - switch positions • 1-8
- Restoring LIBRARY and OPTIONAL save set
 - variations • 1-19
- Run-Time Library • 2-16

S

- Screen management routine
 - installing for VAXTPU • 2-7
- Section file
 - for VAXTPU
 - default file type • 2-2
 - installing • 2-7
 - invoking • 2-7
 - rebuilding • 2-2
- Security • 2-5
- SET QUEUE command • 2-5
 - /ENTRY qualifier • 3-32
 - /NOHOLD qualifier • 3-32
- Shareable image
 - installing VAXTPU • 2-7
- SHOW CLUSTER command • 2-4
- Show Cluster Utility (SHOW CLUSTER) • 2-4
- SHUTDOWN.COM command procedure • 3-14
- Sort/Merge Utility (SORT)
 - changes • 2-15
- Space requirement
 - for decompressing library files • 1-22
- START/CPU command • 3-14
- START/QUEUE/MANAGER command • 2-5
- START/QUEUE command • 2-5
- Stock
 - specifying for a queue • 2-5
- Swap file size
 - use in making alternate root • 1-30
- SYSS\$MANAGER:VMSIMAGES.DAT
 - use in deleting redundant files • 1-23
- SYSTARTUP.COM • 3-15
- System Dump Analyzer (SDA) • 2-23
- System Generation Utility (SYSGEN) • 2-11, 3-30
 - /REMOTE qualifier • 2-11

- System Generation Utility (SYSGEN) (cont'd.)
 - UDABURSTRATE • 3-30
- System management
 - image activation • 3-29
 - INSTALL/GLOBAL/SUMMARY • 3-16
 - LIST/GLOBAL/FULL • 3-16
 - System Generation Utility • 3-30
- System security • 2-5, 3-38
 - password change requirement • 1-13, 1-16
- System service • 2-15
- System upgrade
 - installation summary • 1-4

T

- Time of Day Register (TODR)
 - documentation • 3-50
- to GET_INFO • 3-44
- TPU\$CCTSHR.EXE shareable image • 2-7
- TPU\$COMMAND_TABLE global symbol • 2-2
- TPU\$FACILITY_NAME global symbol • 2-2
- TPU\$MESSAGE_FLAGS global symbol • 2-2
- TPUSECINI logical name • 2-7
- TPUSHR.EXE shareable image • 2-7

U

- UBA (UNIBUS Adapter) • 3-11
- UNIBUS device • 3-11
- Update, mandatory • 1-21
- Upgrade
 - clusters • 1-23
 - materials needed • 1-3
 - notes • 1-3
 - phase 1
 - VAX-11/725, VAX-11/730, and VAX-11/780 Systems • 1-18
 - VAX-11/750 • 1-13
 - VAX-8200/8300 • 1-16
 - phase 2 • 1-19
 - phase 3 • 1-20
 - phase 4 • 1-20
 - phase 5 • 1-20
 - preparation procedure • 1-5
 - restrictions • 1-1
 - single system • 1-4
- Upgrade/optional product
 - installation summary • 1-4

Index

Upgrade procedure

system page file size requirement • 1-7

User-Created Quorum File Problem • 3-31

Utility

Mail Utility in a mixed cluster • 3-3

V

VAX-11/782

BOOTBLDR.COM • 3-6

bootstrap command procedure • 3-6

floppy diskettes

building • 3-6

memory configuration

setting of registers • 3-6

reset memory procedure (RMEM.COM) • 3-6

VAXcluster

alias node identifier • 2-3

node name • 2-3

VAX RMS (Record Management Service)

Image activation • 3-29

VAX Text Processing Utility

See VAXTPU

VAXTPU (VAX Text Processing Utility) • 2-2, 2-7,
3-44

callable interface • 2-2

changing the default editing interface • 2-7

installing • 2-7

packaging • 2-7

pre-defined procedures in EVE • 2-2

section files

default file type • 2-2

installing • 2-7

rebuilding • 2-2

Verify Utility (VERIFY) • 3-16

VMSINSTAL procedure • 3-16

READER'S COMMENTS

Note: This form is for document comments only. DIGITAL will use comments submitted on this form at the company's discretion. If you require a written reply and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Did you find this manual understandable, usable, and well organized? Please make suggestions for improvement.

Did you find errors in this manual? If so, specify the error and the page number.

Please indicate the type of user/reader that you most nearly represent:

- Assembly language programmer
- Higher-level language programmer
- Occasional programmer (experienced)
- User with little programming experience
- Student programmer
- Other (please specify) _____

Name _____ Date _____

Organization _____

Street _____

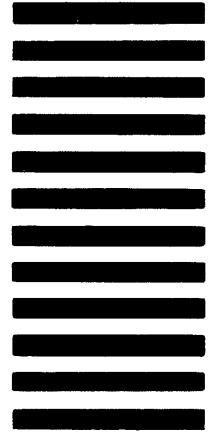
City _____ State _____ Zip Code _____
or Country

Do Not Tear - Fold Here and Tape

digital



No Postage
Necessary
if Mailed in the
United States



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

POSTAGE WILL BE PAID BY ADDRESSEE

SSG PUBLICATIONS ZK1-3/J35
DIGITAL EQUIPMENT CORPORATION
110 SPIT BROOK ROAD
NASHUA, NEW HAMPSHIRE 03062-2698



Do Not Tear - Fold Here

Cut Along Dotted Line

READER'S COMMENTS

Note: This form is for document comments only. DIGITAL will use comments submitted on this form at the company's discretion. If you require a written reply and are eligible to receive one under Software Performance Report (SPR) service, submit your comments on an SPR form.

Did you find this manual understandable, usable, and well organized? Please make suggestions for improvement.

Did you find errors in this manual? If so, specify the error and the page number.

Please indicate the type of user/reader that you most nearly represent:

- Assembly language programmer
- Higher-level language programmer
- Occasional programmer (experienced)
- User with little programming experience
- Student programmer
- Other (please specify) _____

Name _____ Date _____

Organization _____

Street _____

City _____ State _____ Zip Code _____
or Country

Do Not Tear - Fold Here and Tape

digital



No Postage
Necessary
if Mailed in the
United States



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

POSTAGE WILL BE PAID BY ADDRESSEE

SSG PUBLICATIONS ZK1-3/J35
DIGITAL EQUIPMENT CORPORATION
110 SPIT BROOK ROAD
NASHUA, NEW HAMPSHIRE 03062-2698



Do Not Tear - Fold Here

Cut Along Dotted Line