OpenVMS Version 7.0 Release Notes

Order Number: AA–QSBTA-TE

December 1995

This manual describes changes to the software; installation, upgrade, and compatibility information; new and existing software problems and restrictions; and software and documentation corrections.

Revision/Update Information: This is a new manual.

Software Version: OpenVMS VAX Version 7.0
OpenVMS Alpha Version 7.0

Digital Equipment Corporation
Maynard, Massachusetts

December 1995

Digital Equipment Corporation makes no representations that the use of its products in the manner described in this publication will not infringe on existing or future patent rights, nor do the descriptions contained in this publication imply the granting of licenses to make, use, or sell equipment or software in accordance with the description.

Possession, use, or copying of the software described in this publication is authorized only pursuant to a valid written license from Digital or an authorized sublicensor.

Digital conducts its business in a manner that conserves the environment and protects the safety and health of its employees, customers, and the community.

ZK6463

This document is available on CD–ROM.

Contents

Preface xvii

1 OpenVMS Installation, Upgrade, and Hardware Release Notes

1.1 Which Systems Should You Upgrade to OpenVMS Version 7.0? 1–1
1.2 Installation and Upgrade Information Common to VAX and Alpha 1–3
1.2.1 Changes and Enhancements 1–3
1.2.1.1 Software Product Descriptions (SPDs) 1–3
1.2.1.2 Using the Software Products Library to Update InfoServer Software 1–3
1.2.2 Problems and Restrictions 1–4
1.2.2.1 Digital Network Product Requirements 1–4
1.2.2.2 DECnet/OSI: NET$MANAGE Rights Identifier Recommended . . . 1–4
1.2.2.3 InfoServer Client Can Fail to Start If DECnet Is Started or Stopped 1–4
1.2.2.4 POLYCENTER Software Installation Utility Can Hang During Layered Product Installations 1–4
1.2.2.5 VMSINSTAL Can Fail to Update DCL Help During Layered Product Installations 1–5
1.3 Installation and Upgrade Information Specific to VAX 1–5
1.3.1 Problems and Restrictions 1–5
1.3.1.1 Upgrade Version Requirements 1–5
1.3.1.2 MACRO32.EXE and Standalone BACKUP 1–6
1.3.1.3 DECpresent Dependencies for Installing on OpenVMS VAX Version 6.1 or Later 1–6
1.3.1.4 VAXstation and MicroVAX Installation Workaround 1–6
1.4 Installation and Upgrade Information Specific to Alpha 1–6
1.4.1 Changes and Enhancements 1–6
1.4.1.1 Installing the Instruction Emulator 1–6
1.4.1.2 Removing the OpenVMS Alpha Operating System 1–7
1.4.2 Problems and Restrictions 1–8
1.4.2.1 DECram Version 2.2 is Not Supported 1–8
1.4.2.2 RAID Devices Naming Problem 1–9
1.4.2.3 Booting Messages 1–9
1.4.2.4 Using DECnet on Token Ring Devices 1–9
1.5 AlphaServer 2100 Systems (Alpha Only) 1–10
1.5.1 Problems and Restrictions 1–10
1.5.1.1 SCSI Controller Restriction 1–10
1.6 DEC 3000 Series Workstations (Alpha Only) 1–10
1.6.1 Changes and Enhancements 1–10
1.6.1.1 PXG Graphics Board Is No Longer Supported 1–10
1.7 Ethernet and FDDI Controllers 1–10

1.7.1 Problems and Restrictions 1–10
1.7.1.1 Multiple Controllers Configured But Not All Attached to Media (Alpha Only) 1–10
1.8 RF73 and Other RF*nn* DSSI Disk Devices 1–12
1.8.1 Problems and Restrictions 1–12
1.8.1.1 RF73 and Other RF*nn* DSSI Disk Devices and Controller Memory Errors 1–12
1.9 LAT Support for DECservers 1–14
1.9.1 Changes and Enhancements 1–14
1.9.1.1 Increased Baud Rate Support on LAT 1–14

2 General User Release Notes

2.1 DCL Commands 2–1
2.1.1 Problems and Restrictions 2–1
2.1.1.1 SET PROCESS/NOAUTO_UNSHELVE Command in Cluster Environment 2–1
2.1.1.2 SET PROCESS/SUSPEND=KERNEL/ID= Command in Cluster Environment 2–1
2.2 DECforms 2–1
2.2.1 Problems and Restrictions 2–1
2.2.1.1 DECforms Support on OpenVMS Version 7.0 (Alpha Only) 2–2
2.3 DECnet Layered Product 2–2
2.3.1 Changes and Enhancements 2–2
2.3.1.1 DECnet Phase IV Superseded by DECnet/OSI 2–2
2.4 DECTPU 2–2
2.4.1 Problems and Restrictions 2–2
2.4.1.1 Motif Widget Context Help Built-In 2–3
2.5 DECwindows Motif for OpenVMS 2–3
2.5.1 Changes and Enhancements 2–3
2.5.1.1 DECwindows Motif Version 1.1 for OpenVMS No Longer Supported 2–3
2.5.1.2 NCSA Mosaic Browser: TCP/IP Support Options 2–3
2.5.2 Problems and Restrictions 2–3
2.5.2.1 DEC Notes Fails With LinkWorks Error Message (Alpha Only) . . 2–3
2.6 High-Performance Sort/Merge Utility (Alpha Only) 2–4
2.6.1 Problems and Restrictions 2–4
2.6.1.1 Default File Specification 2–4
2.6.1.2 Error Messages 2–4
2.6.1.3 Concurrent Sort Operations 2–4
2.6.1.4 Merging Stream Files Limitation 2–4
2.7 Layered Product Support 2–5
2.8 Mail Utility 2–5
2.8.1 Changes and Enhancements 2–5
2.8.1.1 Specifying Transports 2–5
2.8.1.2 SET FORWARD Command: Specifying Quotation Marks 2–6
2.8.2 Problems and Restrictions 2–6
2.8.2.1 Replying to an Address Containing Nested Quotation Marks 2–6

3 System Management Release Notes

3.1 Audit Analysis Utility 3–1
3.1.1 Documentation Changes and Corrections 3–1
3.1.1.1 *OpenVMS System Management Utilities Reference Manual: A–L* 3–1
3.2 Authorize Utility 3–2
3.2.1 Changes and Enhancements 3–2
3.2.1.1 PQL_MWSEXTENT Can Override WSEXTENT 3–2
3.3 AUTOGEN Command Procedure 3–2
3.3.1 Changes and Enhancements 3–2
3.3.1.1 AGEN$PARAMS.REPORT: Turning Off Logging of DCL Statements 3–2
3.3.1.2 Computation of PQL_DPGFLQUOTA and PQL_MPGFLQUOTA (VAX Only) 3–3
3.3.1.3 NPAGEDYN and NPAGEVIR Limitations and Warnings (VAX Only) 3–3
3.3.1.4 Calculating the Page File Size (Alpha Only) 3–3
3.3.1.5 Paging File Size Computation (VAX Only) 3–4
3.3.1.6 VIRTUALPAGECNT Computations (VAX Only) 3–4
3.3.1.7 WSMAX Computations 3–4
3.3.2 Problems and Restrictions 3–4
3.3.2.1 Full Memory Dumps (Alpha Only) 3–5
3.4 Backup Utility 3–5
3.4.1 Problems and Restrictions 3–5
3.4.1.1 Image and Incremental Backups 3–5
3.4.1.2 Incremental Backups Using PATHWORKS for OpenVMS Servers 3–5
3.4.1.3 Warning Issued on ANALYZE/DISK Operation 3–6
3.4.1.4 CD–ROM Menu System (VAX Only) 3–6
3.4.1.5 Image Backups from an RF73 Disk 3–7
3.4.1.6 Relative File Version -0 Processed Like 0 3–7
3.4.1.7 Standalone BACKUP Version 5.3 on Console Media (VAX Only) 3–7
3.4.1.8 VMS$COMMON.DIR File: Restore Problems 3–8
3.5 Batch and Print Queue Management 3–9
3.5.1 Changes and Enhancements 3–9
3.5.1.1 Process Rights Identifier Limit Raised to 1024 3–9
3.5.2 Problems and Restrictions 3–9
3.5.2.1 PRINT/DELETE Command 3–9
3.5.2.2 SUBMIT/DELETE Command 3–9
3.6 Compatibility Between VAX and Alpha Systems 3–9
3.6.1 Documentation Changes and Corrections 3–9
3.6.1.1 New Manuals Replace Old Ones 3–10
3.7 DCL Commands 3–10
3.7.1 Changes and Enhancements 3–10
3.7.1.1 SHOW NETWORK Command: New Behavior 3–10
3.7.1.2 ANALYZE/ERROR_LOG/SUMMARY Command 3–10
3.7.2 Documentation Changes and Corrections 3–11
3.7.2.1 *OpenVMS DCL Dictionary* 3–11
3.8 Debugger 3–12

3.8.1 Problems and Restrictions 3–12
3.8.1.1 DECwindows Motif Requirement 3–12
3.8.1.2 Displaying a Debug Session from a Personal Computer (PC) 3–12
3.8.1.3 Resource File for Customization and Internationalization 3–12
3.8.1.4 System Resources 3–13
3.8.1.5 User Quotas 3–13
3.9 DECamds 3–13
3.9.1 Changes and Enhancements 3–13
3.9.1.1 Old VAX Versions Not Supported (VAX Only) 3–13
3.9.1.2 New Error Message 3–13
3.9.2 Problems and Restrictions 3–14
3.9.2.1 DECwindows Motif Support 3–14
3.9.2.2 Kernel Threads Not Supported (Alpha Only) 3–14
3.9.3 Corrections 3–14
3.9.3.1 Unique Identification of Disks in a VMScluster 3–14
3.9.3.2 Image Name in Single Process Summary 3–14
3.9.3.3 Saving Resource Hash Table Dense/Sparse Thresholds 3–14
3.10 DECdtm Services in a DECnet/OSI Network 3–14
3.10.1 Problems and Restrictions 3–15
3.10.1.1 Assigning SCSNODE Values 3–15
3.11 DECevent Fault Management Support (Alpha Only) 3–17
3.11.1 Problems and Restrictions 3–17
3.11.1.1 Bit-to-Text Translation Support 3–17
3.11.1.2 Logical File Names 3–17
3.11.2 Documentation Changes and Corrections 3–17
3.11.2.1 Documentation Location 3–17
3.12 DECnet for OpenVMS Alpha (Phase IV) 3–17
3.12.1 Problems and Restrictions 3–17
3.13 DECnet for OpenVMS VAX (Phase IV) 3–18
3.13.1 Changes and Enhancements 3–18
3.13.1.1 DEFQA Now Can Be Configured with SYS$MANAGER:NETCONFIG.COM 3–18
3.13.2 Problems and Restrictions 3–18
3.13.2.1 DEFTA Cannot Be Configured with SYS$MANAGER:NETCONFIG.COM 3–18
3.14 DECnet/OSI for OpenVMS Alpha (Phase V) 3–19
3.14.1 Problems and Restrictions 3–19
3.14.1.1 MOP Incompatible With Kernel Threads 3–19
3.15 DECnet/OSI for OpenVMS VAX (Phase V) 3–19
3.15.1 Changes and Enhancements 3–19
3.15.1.1 DEFQA Now Can Be Configured with SYS$MANAGER:NET$CONFIGURE.COM 3–19
3.15.1.2 DEFTA Now Can Be Configured with SYS$MANAGER:NET$CONFIGURE.COM 3–19
3.16 DECwindows Motif for OpenVMS 3–20
3.16.1 Changes and Enhancements 3–20
3.16.1.1 DECwindows Motif Version 1.1 for OpenVMS No Longer Supported 3–20
3.16.1.2 NCSA Mosaic Browser: TCP/IP Support Options 3–20

3.16.2 Problems and Restrictions 3–20
3.16.2.1 DECwindows Motif Version 1.2 Compatibility Problem With Version 7.0 3–20
3.16.2.2 DEC Notes Fails With LinkWorks Error Message (Alpha Only) . . 3–20
3.16.2.3 Console Broadcasts Disabled 3–21
3.16.2.4 System Files Purged During Startup 3–21
3.16.2.5 Virtual Data Error Message (Alpha Only) 3–21
3.16.2.6 World Wide Support Requires Remedial Kit 3–22
3.16.2.7 DECwindows Server Height or Width Exceeding 32767 (VAX Only) 3–22
3.16.2.8 SET DISPLAY Used to Create WSA Pseudo Workstation Devices 3–23
3.17 Digital Distributed Computing Environment (DCE) for OpenVMS 3–23
3.17.1 Problems and Restrictions 3–23
3.17.1.1 Support Restrictions 3–23
3.18 Help Message Utility (MSGHLP) 3–24
3.18.1 Problems and Restrictions 3–24
3.18.1.1 User-Supplied Comments Are Not Preserved 3–24
3.19 Image Registration 3–24
3.19.1 Problems and Restrictions 3–24
3.19.1.1 Shareable Images Must Be Installed With /OPEN (Alpha Only) 3–24
3.20 INITIALIZE 3–24
3.20.1 Problems and Restrictions 3–24
3.20.1.1 Recovery of Badly Initialized Root Directory (Alpha Only) 3–24
3.21 Install Utility 3–25
3.21.1 Changes and Enhancements 3–25
3.21.1.1 Installing Images 3–25
3.22 Mail Utility 3–25
3.22.1 Changes and Enhancements 3–25
3.22.1.1 MAIL.EXE and Privileges 3–25
3.22.2 Problems and Restrictions 3–26
3.22.2.1 Replying to an Address Containing Nested Quotation Marks 3–26
3.23 Monitoring Performance History (MPH) 3–26
3.24 Monitor Utility 3–28
3.24.1 Changes and Enhancements 3–28
3.24.1.1 TCP/IP Support 3–28
3.24.1.2 Disk Limit Increased 3–28
3.24.1.3 Display Header Changed From "VAX/VMS" To "OpenVMS" (VAX Only) 3–28
3.25 Mount Utility 3–28
3.25.1 Problems and Restrictions 3–28
3.25.1.1 MOUNT/FOREIGN Limitation 3–29
3.25.1.2 ISO 9660 Support in Mixed-Version VMScluster Systems 3–29
3.25.2 Corrections 3–29
3.25.2.1 Noncompliant ISO 9660 CD–ROMs and MOUNT (VAX Only) 3–29
3.25.3 Documentation Changes and Corrections 3–29
3.25.3.1 *OpenVMS System Manager's Manual: Essentials* 3–29
3.26 Nonpaged Pool 3–30
3.26.1 Problems and Restrictions 3–30
3.26.1.1 Prepopulation of Nonpaged Pool (VAX Only) 3–30
3.27 Operator Communication Manager 3–30
3.27.1 Problems and Restrictions 3–30
3.27.1.1 OPERATOR.LOG File 3–31

3.28 POLYCENTER Software Installation Utility 3–31
3.28.1 Problems and Restrictions 3–31
3.28.1.1 PRODUCT Command 3–31
3.28.1.2 Deleting Directories Created by the POLYCENTER Software Installation Utility 3–31
3.28.1.3 Inaccurate Disk Space Reporting 3–32
3.28.1.4 Product Removal Restrictions 3–32
3.29 POLYCENTER Striping (VAX Only) 3–32
3.29.1 Changes and Enhancements 3–32
3.29.1.1 StorageWorks RAID Software for OpenVMS Replaces POLYCENTER Striping 3–32
3.30 POSIX for OpenVMS 3–32
3.30.1 Changes and Enhancements 3–32
3.30.1.1 Ordering POSIX for OpenVMS Version 3.0 3–32
3.30.2 Problems and Restrictions 3–32
3.30.2.1 POSIX for OpenVMS Version 2.0 Is Not Supported 3–32
3.31 RMS Journaling 3–33
3.31.1 Problems and Restrictions 3–33
3.31.1.1 Remote Access of Recovery Unit Journaled Files 3–33
3.31.1.2 VFC Format Sequential Files 3–33
3.32 Security Auditing 3–33
3.32.1 Documentation Changes and Corrections 3–33
3.32.1.1 *OpenVMS System Manager's Manual: Tuning, Monitoring, and Complex Systems* 3–33
3.33 Shared Linkage Sections 3–34
3.33.1 Problems and Restrictions (Alpha Only) 3–34
3.34 SORT32 Work Files 3–34
3.34.1 Changes and Enhancements 3–34
3.34.1.1 Files Now Have Directory Entries 3–34
3.35 StorageWorks RAID Array 110 Subsystem (VAX Only) 3–35
3.35.1 Problems and Restrictions 3–35
3.35.1.1 SHOW DEVICE Does Not Display Capacity 3–35
3.35.1.2 TURBOchannel Devices: Booting Not Supported 3–35
3.36 System Management Utility 3–35
3.36.1 Changes and Enhancements 3–35
3.36.1.1 ALF Command 3–35
3.36.1.2 Rights Identifiers 3–37
3.36.2 Problems and Restrictions 3–38
3.36.2.1 Clusterwide DISKQUOTA Commands (Alpha Only) 3–38
3.37 System Parameters 3–39
3.37.1 Documentation Changes and Corrections 3–39
3.37.1.1 Changes to Online Help 3–39
3.38 System Time 3–40
3.38.1 Changes and Enhancements 3–40
3.38.1.1 Setting Time Using the Battery-Backed Watch (BBW) (Alpha Only) 3–40
3.38.2 Documentation Changes and Corrections 3–40
3.38.2.1 *OpenVMS System Manager's Manual: Essentials* 3–40
3.39 Terminal Fallback Facility (TFF) (Alpha Only) 3–41
3.40 UETP (User Environment Test Package) 3–42
3.40.1 Problems and Restrictions 3–42
3.40.1.1 RRD45 CD–ROM Testing 3–42
3.41 VAX 7000 Systems (VAX Only) 3–43

3.41.1 Problems and Restrictions 3–43
3.41.1.1 Dump Off System Disk (DOSD) Not Supported 3–43
3.42 VAX System Dump Analyzer (SDA) (VAX Only) 3–43
3.42.1 Changes and Enhancements 3–43
3.42.1.1 Dump File Process 3–43
3.42.2 Problems and Restrictions 3–43
3.42.2.1 EXAMINE Command 3–43
3.43 VMScluster Systems 3–43
3.43.1 Changes and Enhancements 3–43
3.43.1.1 SCSI VMScluster Systems 3–43
3.43.1.2 Concurrency Improvements 3–44
3.43.1.3 FDDI Clusters (Alpha Only) 3–44
3.43.2 Problems and Restrictions 3–44
3.43.2.1 VMScluster Tape Serving Requires Adding TMSCP Parameter to MODPARAMS.DAT 3–44
3.43.2.2 KFESA/KFESB Controller Restrictions 3–45
3.43.2.3 System Startup in a VMScluster Environment (Alpha Only) 3–45
3.44 Volume Shadowing 3–45
3.44.1 Changes and Enhancements 3–45
3.44.1.1 Timer-Based Polling 3–46
3.44.1.2 Volume Shadowing Locally Connected SCSI Disks (VAX Only) . . . 3–46
3.44.2 Problems and Restrictions 3–46
3.44.2.1 Failure to Write Crash Dump after Removal of Boot Member (Alpha Only) 3–46
3.44.2.2 SHADZEROMBR Bugcheck for Synchronization Problem 3–47
3.44.2.3 KDM70 Devices 3–47
3.44.2.4 DISMOUNT/CLUSTER Command Can Cause Clusterwide Hang 3–48
3.44.2.5 RZ57 Support Restricted in Shadow Sets 3–48
3.44.3 Documentation Changes and Corrections 3–48
3.44.3.1 *Volume Shadowing for OpenVMS* Manual 3–48

4 Programming Release Notes

4.1 Batch and Print Queues 4–1
4.1.1 Problems and Restrictions 4–1
4.1.1.1 Terminating Executing Batch Jobs 4–1
4.2 Debugger 4–2
4.2.1 Changes and Enhancements 4–2
4.2.1.1 Depositing Values in Arrays: New Error Message 4–2
4.2.2 Problems and Restrictions 4–2
4.2.2.1 COBOL INITIALIZE Statement and Large Tables (Arrays) (Alpha Only) 4–2
4.2.2.2 EXAMINE Command with Large Arrays 4–3
4.2.2.3 Heap Analyzer and /NODEBUG (Alpha Only) 4–3
4.2.2.4 DEBUG.EXE Image (VAX Only) 4–4
4.2.2.5 Watchpoint Support 4–4
4.2.2.6 SHOW CALLS Command (Alpha Only) 4–5
4.2.2.7 DECwindows Motif Interface 4–5
4.2.2.8 EXAMINE LABEL[*n*] Command (Alpha Only) 4–7
4.2.2.9 Kept Debugger 4–7
4.2.2.10 Complex Variables in Fortran Programs (Alpha Only) 4–8
4.2.2.11 Inlined Routines 4–8

4.2.2.12 STEP/INTO Command and User Exception Handlers (Alpha Only) 4–8
4.2.2.13 STEP/OVER Command Error with One-Line Program Loops (Alpha Only) 4–8
4.2.2.14 Vector Support (VAX Only) 4–9
4.2.2.15 Improper Display of Long Text Strings 4–9
4.2.3 Corrections 4–9
4.2.3.1 Ada Boolean Expressions 4–9
4.2.3.2 Ada Enumeration Variables 4–9
4.2.3.3 Arrays: Variable Used as Index 4–9
4.2.3.4 COBOL INITIALIZE Statement and Large Tables (Arrays) (VAX Only) 4–10
4.2.3.5 COBOL Module with the Same Name as a Variable 4–10
4.2.3.6 COBOL Scaled Computational Data (Alpha Only) 4–10
4.2.3.7 Customizations of Windows Interface 4–10
4.2.3.8 %DEBUG-E-LINEINFO Message Information on Line Numbers 4–10
4.2.3.9 DEPOSIT Command with Large Unsigned Integer 4–10
4.2.3.10 EVALUATE Command and Truncation Problem (Alpha Only) . . . 4–10
4.2.3.11 Fortran Complex Numbers (Alpha Only) 4–10
4.2.3.12 Heap Analyzer and Memory 4–11
4.2.3.13 Library Text Modules and SEARCH Command 4–11
4.2.3.14 Null Frame Procedures and SHOW CALLS (Alpha Only) 4–11
4.2.3.15 SET WATCH, STEP, and GO Commands (Alpha Only) 4–11
4.2.3.16 Incorrect Source Code Displayed 4–11
4.2.3.17 Incomplete Program State Information 4–11
4.2.3.18 Inaccurate Displays in Multiprocess Applications 4–11
4.3 DEC Ada Run-Time Library 4–11
4.3.1 Problems and Restrictions 4–11
4.3.1.1 Unexpected Storage Errors (Alpha Only) 4–12
4.4 DEC C++ 4–12
4.4.1 Problems and Restrictions 4–12
4.4.1.1 SYS$LIB_C.TLB Structures Need Prototypes (Alpha Only) 4–12
4.5 DEC C++ Class Library 4–12
4.5.1 Changes and Enhancements 4–12
4.5.1.1 Threadsafe Support 4–12
4.5.2 Corrections 4–13
4.6 DEC C Run-Time Library 4–13
4.6.1 Changes and Enhancements 4–13
4.6.1.1 Mapping of Record Carriage Control Data 4–13
4.6.1.2 Open File Limit Increased 4–14
4.6.1.3 Support Increased for Number of Open Sockets 4–14
4.6.1.4 New DEC C RTL Functions 4–14
4.6.1.5 ICONV Utility Enhancements 4–15
4.6.1.6 Specifying All Versions of File in remove( ) 4–16
4.6.1.7 Date and Time Support Changes 4–16
4.6.1.8 Message Specifications Relaxed 4–17
4.6.1.9 ISO Amendment 1 Changes to fscanf( ) Functions 4–17
4.6.1.10 New 4.4 BSD Curses Functions (Alpha Only) 4–18
4.6.1.11 Internationalization Support 4–18
4.6.1.12 LOCALE Utility Change 4–18
4.6.1.13 Floating-Point Values Now Conform to IEEE Standard 4–18
4.6.1.14 Math Functions Conform to XPG4 (VAX Only) 4–19
4.6.1.15 Open Routines Now Validate RMS Options 4–19

4.6.1.16 Setvbuf Function 4–19
4.6.1.17 Sleep Function Now Returns Unslept Seconds 4–19
4.6.1.18 Wildcards No Longer Accepted by stat Function 4–19
4.6.1.19 Writing to Pipes Now Conforms to POSIX Standards 4–20
4.6.2 Problems and Restrictions 4–20
4.6.2.1 Socket Behavior Prior to UCX V3.3 4–20
4.6.2.2 Linking DEC C Applications Using /NOSYSSHR 4–20
4.6.2.3 Linking With /NOSYSSHR Disables Dynamic Activation 4–20
4.6.2.4 Internationalization Compatibility Problem with DECwindows Motif 4–20
4.6.3 Corrections 4–21
4.6.3.1 Open Sockets Passed to Child Processes 4–21
4.6.3.2 File Protection Status Now Mapped to EACCES 4–21
4.6.3.3 Invalid RMS Options Cause Open to Fail 4–21
4.6.3.4 LOCALEDEF Utility 4–21
4.6.3.5 UNIX File Specifications Containing Relative Paths 4–21
4.6.3.6 Behavior of realloc( ) Function 4–22
4.6.3.7 Usage of SYS_ERRLIST and SYS_NERR 4–22
4.6.3.7.1 Recompiling with DEC C Version 4.1 or Version 5.0 (Alpha Only) 4–23
4.6.3.7.2 Recompiling Pre-Version 4.1 (OpenVMS Alpha) or Pre-Version 5.0 (OpenVMS VAX) DEC C Compilers 4–24
4.6.3.7.3 Relinking Programs Compiled on Earlier Versions of OpenVMS 4–25
4.7 DECthreads 4–25
4.7.1 Changes and Enhancements 4–25
4.7.1.1 POSIX 1003.1c Standard Style Interface 4–25
4.7.1.2 Thread Independent Services (TIS) Interface 4–26
4.7.1.3 POSIX 1003.4a Draft 4 Interface Retirement 4–26
4.7.1.4 CMA Interface: Change in Status 4–26
4.7.1.5 Application Coding Errors 4–26
4.7.1.6 Timed Condition Wait Time Models 4–26
4.7.1.7 $EXIT and Exit Handling Changes (Alpha Only) 4–27
4.7.1.8 Condition Variable Waits (Alpha Only) 4–27
4.7.1.9 Behavior on Multiprocessors (Alpha Only) 4–27
4.7.1.10 Blocking System Service Calls (Alpha Only) 4–28
4.7.1.11 $HIBER (Alpha Only) 4–28
4.7.1.12 DECthreads Debugger 4–28
4.7.2 Problems and Restrictions 4–28
4.7.2.1 Release Compatibility 4–28
4.7.2.2 Language Support 4–28
4.7.2.3 64-Bit Exception Handling (Alpha Only) 4–29
4.7.2.4 64-Bit Parameter Support in Routines (Alpha Only) 4–29
4.7.2.5 Using Multiple Kernel Threads on Debugger (Alpha Only) 4–29
4.7.2.6 Interoperability with POSIX for OpenVMS 4–29
4.7.2.7 DECthreads Debugger Metering Function 4–29
4.7.2.8 Condition Variable Attributes Routines 4–29
4.7.2.9 C Run-Time Library *errno* Value 4–29
4.7.2.10 SET TASK/ACTIVE Command 4–29
4.7.2.11 Dynamic Image Activation 4–30
4.7.2.12 Exit Handler Routine (VAX Only) 4–30
4.7.2.13 DECthreads Exit Routines Can Terminate Process (VAX Only) . . . 4–30
4.7.3 Documentation Changes and Corrections 4–31
4.7.3.1 Online Help 4–31

4.8 DECTPU for DECwindows Motif 4–31
4.8.1 Problems and Restrictions 4–31
4.8.1.1 Small Display Monitors and DECwindows Motif Applications 4–31
4.9 Delta/XDelta Debugger 4–31
4.9.1 Changes and Enhancements 4–31
4.9.1.1 Image Activation (Alpha Only) 4–31
4.9.2 Documentation Changes and Corrections 4–32
4.9.2.1 *OpenVMS Delta/XDelta Debugger Manual* 4–32
4.9.2.1.1 Floating Point Register Support (Alpha Only) 4–32
4.9.2.1.2 Base Register Default Offset (Alpha Only) 4–32
4.10 Device Support on OpenVMS Alpha Systems 4–32
4.10.1 Changes and Enhancements 4–32
4.10.1.1 OpenVMS Alpha Device Drivers Must Be Recompiled and Relinked 4–32
4.10.1.2 Bus Array Entry Structure Change 4–33
4.10.1.3 SYS$MSBDRIVER Removed from OpenVMS Distribution 4–33
4.10.1.4 Enhanced SCSI Interface to IO$_DIAGNOSE Function 4–33
4.10.1.5 PCI DMA Window Changes 4–33
4.10.1.6 Device IPL Setup for Drivers 4–35
4.10.1.7 SCSI Port and Class Driver Changes 4–35
4.10.1.7.1 OpenVMS Alpha SCSI Class Driver Changes 4–36
4.10.2 Problems and Restrictions 4–36
4.10.2.1 SYS$MSBDRIVER Load Failure 4–36
4.10.2.2 SCSI Firmware Support 4–37
4.10.2.2.1 SCSI Firmware Update Instructions 4–37
4.10.2.2.2 Firmware Revision Level 442D Requirements 4–37
4.10.2.2.3 442D Firmware Installation Procedure 4–38
4.10.2.3 SCSI Port and Class Drivers 4–38
4.10.2.3.1 Mounting Third-Party SCSI Disks 4–38
4.10.2.3.2 Add-On SCSI Adapters 4–38
4.10.2.3.3 HSZ40 and Transportable SCSI Disk Shadow-Set Members 4–39
4.10.2.3.4 SCSI Disk I/O Performance Degradation for KZMSA XMI and Adaptec 1742A Adapters 4–39
4.10.2.4 C Structure Member Names: Passing to Nested C Macros 4–40
4.10.2.5 OpenVMS Alpha System-Code Debugger 4–40
4.11 Device Support on OpenVMS VAX Systems 4–41
4.11.1 Documentation Changes and Corrections 4–41
4.11.1.1 *OpenVMS VAX Device Support Manual* 4–41
4.11.1.1.1 Linking a Device Driver 4–41
4.11.1.1.2 Device-Register I/O Space: Usage Restrictions 4–41
4.11.1.2 *OpenVMS VAX Device Support Reference Manual* 4–42
4.11.1.2.1 COM$DRVDEALMEM Routine Synchronization 4–42
4.11.1.2.2 CRB Data Structure 4–42
4.11.1.2.3 SCDRP Data Structure SCSI Flags 4–42
4.11.1.2.4 SPI$CONNECT Macro 4–42
4.11.1.2.5 SPI$GET_CONNECTION_CHAR and SPI$SET_CONNECTION_CHAR Macros 4–43
4.11.1.2.6 $EQULST Macro 4–43
4.12 Event Flags 4–43
4.12.1 Changes and Enhancements 4–43
4.12.1.1 Alternative to Local Event Flags 4–43
4.13 High-Performance Sort/Merge Utility (Alpha Only) 4–44

4.13.1 Problems and Restrictions 4–44
4.13.1.1 Default File Specification 4–44
4.13.1.2 Error Messages 4–44
4.13.1.3 Concurrent Sort Operations 4–44
4.13.1.4 Merging Stream Files Limitation 4–44
4.14 I/O User 4–45
4.14.1 Problems and Restrictions 4–45
4.14.1.1 I/O Write Operations Interrupted by System Failure 4–45
4.14.2 Documentation Changes and Corrections 4–46
4.14.2.1 *OpenVMS I/O User's Reference Manual* 4–46
4.15 Librarian Utility 4–46
4.15.1 Problems and Restrictions 4–46
4.15.1.1 PGFLQUOTA Should Exceed 23000 (Alpha Only) 4–46
4.16 Linker Utility 4–47
4.16.1 Problems and Restrictions 4–47
4.16.1.1 Fixup Information Not Generated (Alpha Only) 4–47
4.16.1.2 Shareable Image Psects Pointed to by Symbol Definitions (Alpha Only) 4–47
4.16.2 Documentation Changes and Corrections 4–47
4.16.2.1 *OpenVMS Linker Utility Manual* 4–48
4.17 Lock Manager 4–48
4.17.1 Changes and Enhancements 4–48
4.17.1.1 LCKMGR Spin Lock (Alpha Only) 4–48
4.18 LTDRIVER 4–48
4.18.1 Problems and Restrictions 4–48
4.18.1.1 CANCEL SELECTIVE Cannot Cancel IO$_TTY_PORT Functions 4–48
4.19 MACRO-32 Compiler for OpenVMS Alpha (Alpha Only) 4–48
4.19.1 Problems and Restrictions 4–49
4.19.1.1 .CALL_ENTRY Requires New STARLET.MLB 4–49
4.19.1.2 .ASCID Directive 4–49
4.19.1.3 .GLOBAL_LABEL Directive 4–49
4.19.1.4 Floating-Point Return Values in R0 4–49
4.19.1.5 INSV Instructions Negate Granularity Preservation 4–50
4.19.1.6 .JSB_ENTRY Directive 4–50
4.19.1.7 Floating-Point and Packed-Decimal Instruction Arguments 4–50
4.19.1.8 Multiple Object Files Compilation 4–50
4.19.2 Documentation Changes and Corrections 4–50
4.19.2.1 *Porting VAX MACRO Code from OpenVMS VAX to OpenVMS Alpha* 4–50
4.19.2.1.1 Maintaining Common Sources for VAX and Alpha 4–50
4.19.2.1.2 Packed-Decimal and Floating-Point Instructions and the VAX AP 4–51
4.19.2.1.3 Alignment Considerations for Atomicity: Correction to Example 4–51
4.19.2.1.4 Entry Point Declarations 4–52
4.19.2.1.5 BUG*x* Instruction Is Not Supported 4–52
4.19.2.1.6 /FLAGGING=INSTRUCTIONS Qualifier 4–53
4.19.2.1.7 TRACEBACK and DEBUG Options Restriction Removed 4–53
4.20 Mathematics (MTH$) Run-Time Library 4–53
4.20.1 Problems and Restrictions 4–53
4.20.1.1 Linking Images to Run on Previous OpenVMS VAX Versions (VAX Only) 4–53
4.21 Migration (VAX to Alpha) 4–54

4.21.1 Documentation Changes and Corrections 4–54
4.21.1.1 New Manuals Replace Old Ones 4–54
4.22 POLYCENTER Software Installation Utility 4–54
4.22.1 Problems and Restrictions 4–54
4.22.1.1 Account Removal Problem 4–54
4.22.1.2 Alternate File Placement Not Supported 4–55
4.22.1.3 Conflict Error Reporting 4–55
4.22.1.4 Error Reporting 4–55
4.22.1.5 File Generation Restrictions 4–55
4.22.1.6 Information Statement Problem 4–56
4.22.1.7 Multiple Execute Remove Statements 4–56
4.22.1.8 Multiple Operating Scopes Not Supported 4–56
4.22.1.9 Package Operation Constraints 4–56
4.22.1.10 Partial Kit Restrictions 4–56
4.22.1.11 Rights Identifier Problem 4–57
4.22.1.12 "Uses" Clause File Specification Restriction 4–57
4.22.2 Documentation Changes and Corrections 4–57
4.22.2.1 DCL Help: PRODUCT PACKAGE Command 4–57
4.22.2.2 *POLYCENTER Software Installation Utility Developer's Guide* . . . 4–57
4.22.2.2.1 File Statement 4–57
4.22.2.2.2 Scope Statement 4–57
4.23 Privileged Interfaces and Data Structures (Alpha Only) 4–58
4.24 Run-Time Libraries 4–58
4.24.1 Changes and Enhancements 4–58
4.24.1.1 PPL (Parallel Processing Library) to Be Retired 4–58
4.24.2 Problems and Restrictions 4–58
4.24.2.1 Run-Time Libraries Not Included in OpenVMS Alpha Systems (Alpha Only) 4–59
4.25 Run-Time Library (LIB$) 4–59
4.25.1 Corrections 4–59
4.25.1.1 LIB$SHOW_ZONE and Large Lookaside Lists 4–59
4.26 Screen Management (SMG$) Facility 4–59
4.26.1 Changes and Enhancements 4–59
4.26.1.1 VT500 Series Terminal Support 4–59
4.27 Symbiont/Job Controller Interface (SMB) Routines 4–61
4.27.1 Changes and Enhancements 4–61
4.27.1.1 SMB$READ_MESSAGE_ITEM Routine Has a New Symbol 4–61
4.28 System Services 4–61
4.28.1 Problems and Restrictions 4–61
4.28.1.1 Linking SECURESHR Images to Run on Older Versions 4–61
4.28.1.2 $SUSPND Behaves Incorrectly in a Cluster Environment 4–62
4.29 System Time Operation 4–62
4.29.1 Documentation Changes and Corrections 4–62
4.29.1.1 *OpenVMS Programming Concepts Manual* 4–62
4.30 VAX MACRO (VAX Only) 4–64
4.30.1 Documentation Changes and Corrections 4–64
4.30.1.1 *VAX MACRO and Instruction Set Reference Manual* 4–64
4.30.1.1.1 .ASCIZ Macro Directive 4–64
4.30.1.1.2 General Register Addressing 4–64
4.30.1.1.3 VAX Instruction CASE 4–64
4.31 X/Open Transport Interface (XTI) 4–65
4.31.1 Changes and Enhancements 4–65
4.31.2 Problems and Restrictions 4–66

A Remedial Kits Included in OpenVMS Version 7.0

A.1 Remedial Kits Included in OpenVMS VAX Version 7.0 . . . . A–1
A.2 Remedial Kits Included in OpenVMS Alpha Version 7.0 . . . . A–2

B Remedial Kits Not Included in OpenVMS Version 7.0

B.1 Remedial Kits Not Included in OpenVMS VAX Version 7.0 . . . . B–1
B.1.1 Record Management Services (RMS) Kit (VAXRMS02_061) B–2
B.1.2 System Services Kit (VAXSYS04_062) B–2
B.2 Remedial Kits Not Included in OpenVMS Alpha Version 7.0 . . . . B–2
B.2.1 LAST Control Program (LASTCP) Kit (ALPLAD02_062) B–2
B.2.2 SCSI Driver Kit (ALPSCSI03_061) B–2

Index

Figures

3–1 Transaction Group 3–16

Tables

1–1 Error-Free Microcode Revision Levels 1–12
1–2 Commands for Updating Microcode in Certain DSSI Disk Devices . . . 1–14
3–1 TFF Character Fallback Tables 3–41
4–1 Debugger Commands Not Available in the DECwindows Motif Interface 4–6
4–2 Obsolete SPI Interface Macros 4–36
4–3 Revision Level 442D Firmware Compatibility 4–37
4–4 Supported Terminal Devices 4–46
4–5 Run-Time Libraries Not Included on OpenVMS Alpha Systems 4–59
4–6 Predefined Output Date Formats 4–63
4–7 Predefined Output Time Formats 4–63

Preface

Intended Audience

This manual is intended for all OpenVMS operating system users. Read this manual before you install, upgrade, or use Version 7.0 of the operating system.

Document Structure

This manual contains the following chapters and appendixes:

- Chapter 1 contains release notes that pertain to installing and upgrading the OpenVMS operating systems, as well as hardware-related information.
- Chapter 2 contains release notes that relate to the general use of the OpenVMS operating system.
- Chapter 3 contains release notes specific to system management information.
- Chapter 4 contains release notes that relate to programming on an OpenVMS system.
- Appendix A lists remedial kits that are included in OpenVMS Version 7.0.
- Appendix B describes remedial kits that are *not* included in OpenVMS Version 7.0.

This manual contains release notes introduced in the current release and notes from previous OpenVMS versions that still apply to the new release. Margin notes for each release note indicate the version of origin (for example, *V6.2*).

Notes from previous releases are published when:

- The release note has not been documented in hard copy in any other manual in the OpenVMS documentation set, and the note is still pertinent.
- The release note may be pertinent in multiple-version OpenVMS cluster systems.

Related Documents

For a list of additional documents that are available in support of this version of the OpenVMS operating system, refer to the *Overview of OpenVMS Documentation*.

For additional information about OpenVMS products and services, access the Digital OpenVMS World Wide Web site using the following URL:

```
http://www.openvms.digital.com
```

Reader’s Comments

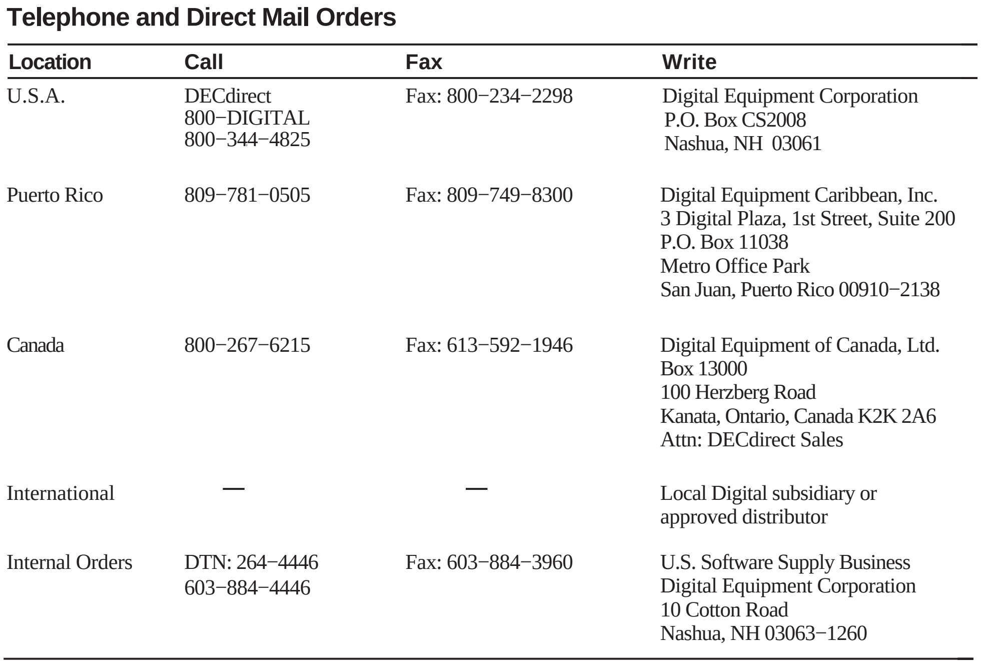
Digital welcomes your comments on this manual.

Print or edit the online form SYS$HELP:OPENVMSDOC_COMMENTS.TXT and send us your comments by:

Internet	**openvmsdoc@zko.mts.dec.com**
Fax	603 881-0120, Attention: OpenVMS Documentation, ZK03-4/U08
Mail	OpenVMS Documentation Group, ZKO3-4/U08 110 Spit Brook Rd. Nashua, NH 03062-2698

How To Order Additional Documentation

Use the following table to order additional documentation or information. If you need help deciding which documentation best meets your needs, call 800-DIGITAL (800-344-4825).



Telephone and Direct Mail Orders

Location	Call	Fax	Write
U.S.A.	DECdirect 800−DIGITAL 800−344−4825	Fax: 800−234−2298	Digital Equipment Corporation P.O. Box CS2008 Nashua, NH 03061
Puerto Rico	809−781−0505	Fax: 809−749−8300	Digital Equipment Caribbean, Inc. 3 Digital Plaza, 1st Street, Suite 200 P.O. Box 11038 Metro Office Park San Juan, Puerto Rico 00910−2138
Canada	800−267−6215	Fax: 613−592−1946	Digital Equipment of Canada, Ltd. Box 13000 100 Herzberg Road Kanata, Ontario, Canada K2K 2A6 Attn: DECdirect Sales
International	—	—	Local Digital subsidiary or approved distributor
Internal Orders	DTN: 264−4446 603−884−4446	Fax: 603−884−3960	U.S. Software Supply Business Digital Equipment Corporation 10 Cotton Road Nashua, NH 03063−1260

ZK−7654A−GE

Conventions

The name of the OpenVMS AXP operating system has been changed to OpenVMS Alpha. Any references to OpenVMS AXP or AXP are synonymous with OpenVMS Alpha or Alpha.

The following conventions are used to identify information specific to OpenVMS Alpha or to OpenVMS VAX:

Alpha	The Alpha icon denotes the beginning of information specific to OpenVMS Alpha.

VAX

The VAX icon denotes the beginning of information specific to OpenVMS VAX.

♦

The diamond symbol denotes the end of a section of information specific to OpenVMS Alpha or to OpenVMS VAX.

In this manual, every use of DECwindows and DECwindows Motif refers to DECwindows Motif for OpenVMS software.

The following conventions are also used in this manual:

Ctrl/*x*

A sequence such as Ctrl/*x* indicates that you must hold down the key labeled Ctrl while you press another key or a pointing device button.

PF1 *x* or GOLD *x*

A sequence such as PF1 *x* or GOLD *x* indicates that you must first press and release the key labeled PF1 or GOLD and then press and release another key or a pointing device button.

GOLD key sequences can also have a slash (/), dash (–), or underscore (_) as a delimiter in EVE commands.

Return

In examples, a key name enclosed in a box indicates that you press a key on the keyboard. (In text, a key name is not enclosed in a box.)

. . .

Horizontal ellipsis points in examples indicate one of the following possibilities:

- Additional optional arguments in a statement have been omitted.
- The preceding item or items can be repeated one or more times.
- Additional parameters, values, or other information can be entered.

.
.
.

Vertical ellipsis points indicate the omission of items from a code example or command format; the items are omitted because they are not important to the topic being discussed.

( )

In command format descriptions, parentheses indicate that, if you choose more than one option, you must enclose the choices in parentheses.

[ ]

In command format descriptions, brackets indicate optional elements. You can choose one, none, or all of the options. (Brackets are not optional, however, in the syntax of a directory name in an OpenVMS file specification or in the syntax of a substring specification in an assignment statement.)

{ }

In command format descriptions, braces indicate a required choice of options; you must choose one of the options listed.

boldface text

Boldface text represents the introduction of a new term or the name of an argument, an attribute, or a reason.

Boldface text is also used to show user input in Bookreader versions of the manual.

italic text

Italic text indicates important information, complete titles of manuals, or variables. Variables include information that varies in system output (Internal error *number*), in command lines (/PRODUCER=*name*), and in command parameters in text (where *device-name* contains up to five alphanumeric characters).

UPPERCASE TEXT

Uppercase text indicates a command, the name of a routine, the name of a file, or the abbreviation for a system privilege.

`Monospace type`

Monospace type indicates code examples and interactive screen displays.

In the C programming language, monospace type in text identifies the following elements: keywords, the names of independently compiled external functions and files, syntax summaries, and references to variables or identifiers introduced in an example.

-

A hyphen at the end of a command format description, command line, or code line indicates that the command or statement continues on the following line.

numbers

All numbers in text are assumed to be decimal unless otherwise noted. Nondecimal radixes—binary, octal, or hexadecimal—are explicitly indicated.

1

OpenVMS Installation, Upgrade, and Hardware Release Notes

This chapter contains information that applies to installations and upgrades of the OpenVMS VAX and OpenVMS Alpha operating systems. It also provides information specific to certain hardware.

The installation and upgrade notes in this chapter are organized into the following categories:

- Things to consider before upgrading to OpenVMS Version 7.0 (see Section 1.1)
- Installation and upgrade notes common to both VAX and Alpha systems (see Section 1.2)
- Alpha specific installation and upgrade notes (see Section 1.4)
- VAX specific installation and upgrade notes (see Section 1.3)

Hardware and firmware notes follow the upgrade and installation sections.

1.1 Which Systems Should You Upgrade to OpenVMS Version 7.0?

OpenVMS Version 6.2 and Version 7.0 ship together on a single operating system kit. Both releases are fully qualified to meet the highest quality standards. Either can be used in a production environment. Both versions will be superseded by the next major Version 7.*n* OpenVMS release.

You must decide which version to use, depending on your environment and application requirements. If you run in a production environment, you might prefer to remain on the Version 6.2–*n* track, whereas if you want to use the new technology as soon as possible or you are an independent software vendor (ISV), you might choose the Version 7.0 track. The advantages of each version are summarized in the following paragraphs.

OpenVMS Version 7.0

OpenVMS Version 7.0 contains many new features, which are described in the *OpenVMS Version 7.0 New Features Manual*. Most of these features, such as 64-bit addressing support and Fast I/O, require modifications to existing, privileged applications. Therefore, Version 7.0 is probably of most interest to users who are developing new applications or modifying existing applications to take advantage of these new features. Major new features of Version 7.0 are as follows:

- Version 7.0 features on both VAX and Alpha:
  - Clusterwide $CREPRC
  - Enhancements for Internet support

Alpha

- Version 7.0 features on Alpha only:
  - 64-bit addressing
  - Kernel threads
  - Fast I/O
  - Fast Path
  - Wind/U
  - Spiralog file system ♦

VAX

- Version 7.0 feature on VAX only:
  - Alpha-compatible libraries to simplify cross-architecture application development ♦

OpenVMS Version 6.2–*n*

New systems that include factory-installed software (FIS) ship with OpenVMS Version 6.2. In response to customer feedback, Digital plans to release several updates of Version 6.2 over the next six to twelve months to support new, faster CPUs and new I/O adapters. Support for these new systems and storage, which do not require modifications to existing applications, will be included only in the Version 6.2–*n* stream — not in Version 7.0. This strategy allows users to take advantage of the latest hardware without upgrading to Version 7.0 or modifying applications. If you want to use the new hardware and can have an ISV update your applications in preparation for the next major Version 7.*n* OpenVMS release, you might wish to stay with the Version 6.2–*n* stream. Major features expected to be included in the Version 6.2–*n* stream are as follows:

- Version 6.2–*n* features on both VAX and Alpha:
  - New systems and storage

Alpha

- Version 6.2–*n* features on Alpha only:
  - Fast Wide Differential (FWD) SCSI clusters
  - PCI-to-CI adapter ♦

Mixed-Version Clusters

While Digital recommends that all nodes in a VMScluster run the same version, running OpenVMS Version 7.0 and Version 6.2–*n* together in the same cluster is supported. A mixed-version cluster may be appropriate if you use some cluster nodes for application development and others for production work.

What’s Next?

The features of both Version 6.2–*n* and Version 7.0 will converge in the next major Version 7.*n* OpenVMS release. Many applications and layered products will be available that take advantage of the new features implemented in Version 7.0 and all hardware supported by the Version 6.2–*n* variants will also be supported. Contact a Digital support representative for information about upcoming Version 6.2–*n* hardware releases over the next 6 to 12 months.

1.2 Installation and Upgrade Information Common to VAX and Alpha

The following notes document installation and upgrade information common to both platforms. For more VAX specific installation and upgrade notes, see Section 1.3. For additional Alpha specific notes, see Section 1.4.

1.2.1 Changes and Enhancements

This section describes information related to installing or upgrading the OpenVMS VAX or OpenVMS Alpha operating system.

1.2.1.1 Software Product Descriptions (SPDs)

V7.0 Printed copies of the Software Product Descriptions (SPDs) are no longer provided. However, the following SPDs are now available on the OpenVMS Version 7.0 media:

- OpenVMS Operating System for Alpha and VAX, Version 7.0
- OpenVMS Cluster Software
- DECnet for OpenVMS, Version 7.0
- RMS Journaling for OpenVMS, Version 7.0
- Volume Shadowing for OpenVMS, Version 7.0

Customers who receive the operating system and documentation on CD–ROM can consult the *Guide to OpenVMS Version 7.0 CD–ROMs* for complete information about locating and printing the SPDs. VAX users who receive their distribution kit on other media should refer to the *OpenVMS VAX Version 7.0 Upgrade and Installation Manual* for information about how to locate and access the SPD files on their kit.

1.2.1.2 Using the Software Products Library to Update InfoServer Software

V6.1 You can use the Software Products Library (formerly known as ConDIST) to update InfoServer software. After you log in to the InfoServer system, perform the following steps:

1. Insert the disk containing the [INFOSERVxxx] directory tree in a compact disk drive attached to the InfoServer system.
2. At the InfoServer> prompt, enter a command in the following format, where *n* is the drive number:
   - On the InfoServer 100 or InfoServer 150 system, enter a command in the following format:

     UPDATE SYSTEM DK*n*:
   - On the InfoServer 1000 system, enter a command in the following format:

     UPDATE SYSTEM DK*n*: FLASH

   These commands move the InfoServer software to the internal read/write device. The next time you boot the InfoServer system, it runs the updated software. Note that you can also boot the server from the Software Products Library disk.

1.2.2 Problems and Restrictions

This section contains upgrade and installation problems and restrictions that are common to both VAX and Alpha operating systems.

1.2.2.1 Digital Network Product Requirements

V6.2

VAX

On OpenVMS VAX systems, if you currently run DECnet/OSI, the kit for OpenVMS VAX Version 6.2 or later detects this and gives you the option to retain those files. If you prefer to use the Phase IV code base, you have the option to do so. However, if you want to use DECnet over TCP/IP, you must run DECnet/OSI. ♦

V7.0

Alpha

On OpenVMS Alpha systems, if DECnet/OSI is currently installed, it is automatically upgraded to Version 7.0 immediately after the OpenVMS Alpha upgrade, as described in the *OpenVMS Alpha Version 7.0 Upgrade and Installation Manual*. If you want to use the DECnet over TCP/IP feature of DECnet/OSI, you must use Version 4.0A of Digital TCP/IP Services for OpenVMS. ♦

1.2.2.2 DECnet/OSI: NET$MANAGE Rights Identifier Recommended

V6.1

After installing the DECnet/OSI for OpenVMS product, Digital recommends that you grant the NET$MANAGE rights identifier to the SYSTEM account or any other account from which you expect to use VMSINSTAL. Installations of some layered products may fail if you are running DECnet/OSI and the account running the installation does not have this identifier.

1.2.2.3 InfoServer Client Can Fail to Start If DECnet Is Started or Stopped

V6.0

The InfoServer client software will fail to start on a system where DECnet has been started and subsequently stopped. The following message will be found in the file SYS$MANAGER:ESS$STARTUP.LOG:

```
%ESS-I-NONET ESS started before DECnet.  4-MAR-1994 16:36:39.29
```

This is caused by a problem in the startup procedure SYS$STARTUP:ESS$STARTUP.COM.

If the InfoServer client must be started at this point, the LASTport transport can be started with the Last Control Program using the following command:

```
$ MCR ESS$LASTCP
LASTCP> START
```

This command will start the transport. You may now execute the InfoServer client startup:

```
$ @SYS$STARTUP:ESS$STARTUP DISK
```

Because the last transport is already started, the startup will run successfully.

1.2.2.4 POLYCENTER Software Installation Utility Can Hang During Layered Product Installations

V6.2

During a layered product installation, if another process is accessing the DCL help library, the POLYCENTER Software Installation utility will hang until it can open the file for exclusive access. When all users have exited from DCL Help, the utility automatically resumes installation of the layered product. This wait condition occurs after the "Portion Done: 0%..." message is displayed but before the percent complete indicator reaches 100%. However, no explicit warning message is displayed to alert the user who issued the PRODUCT INSTALL command that one or more other users have the DCL help library open.

1.2.2.5 VMSINSTAL Can Fail to Update DCL Help During Layered Product Installations

V6.0 During a layered product installation, if another process is accessing DCL Help, the following events occur:

- The installer sees the following message displayed once:

  ```
  %VMSINSTAL-I-DCLHLPINUSE, Trying to update DCL HELP library.  Procedure
  will try three more times.
  ```

  The procedure makes up to three additional attempts to access DCL Help (one attempt every 1 1/2 minutes).

- All user processes see the following message up to three times (that is, each time VMSINSTAL attempts and fails to access DCL Help):

  ```
  Software installation procedure in progress, but DCL HELP command is
  in use. Trying to update DCL HELP library.  Please exit DCL HELP
  command temporarily.
  ```

- After three tries to update DCL Help (4 1/2 minutes), if DCL Help is still accessed, VMSINSTAL does the following:

  1. Moves the files to be updated to a working directory
  2. Notifies the installer with the following message:

     ```
     %VMSINSTAL-I-NODCLHLP, DCL HELP not provided for this product.
     Manually update HELP libraries after installation.
     Use SYS$COMMON:[SYSHLP]<file name> for providing
     new HELP
     ```

  3. At the completion of the installation, issues the following message to the installer:

     ```
     %VMSINSTAL-I-REFDCLHLP, DCL HELP could not be updated.
     Reference SYS$UPDATE:DODCLHELP.VMI for information updating
     DCL HELP.
     ```

1.3 Installation and Upgrade Information Specific to VAX

VAX The release notes in this section pertain only to installations or upgrades of OpenVMS VAX operating systems. See Section 1.2 for additional notes that pertain to both VAX and Alpha systems. For complete information about installing or upgrading your OpenVMS VAX Version 7.0 operating system, refer to the *OpenVMS VAX Version 7.0 Upgrade and Installation Manual*.

1.3.1 Problems and Restrictions

This section describes problems and restrictions that apply to installing or upgrading to OpenVMS VAX Version 7.0.

1.3.1.1 Upgrade Version Requirements

V7.0 To upgrade your system to OpenVMS VAX Version 7.0, you must have OpenVMS VAX Version 6.1 or higher already installed.

To perform a concurrent upgrade or rolling upgrade in a VAXcluster environment, all nodes in the cluster must be running at least OpenVMS VAX Version 6.1. For rolling upgrades, Digital recommends that all nodes in the cluster run the same version of the operating system.

For more information about upgrading to OpenVMS VAX Version 7.0, see the *OpenVMS VAX Version 7.0 Upgrade and Installation Manual*.

1.3.1.2 MACRO32.EXE and Standalone BACKUP

V6.2 To build standalone BACKUP onto tape, you must have the image MACRO32.EXE on your system disk. Do not tailor off the Macro libraries before building standalone BACKUP onto tape.

1.3.1.3 DECpresent Dependencies for Installing on OpenVMS VAX Version 6.1 or Later

V6.1 To run DECpresent Version 1.0A on OpenVMS VAX Version 6.1 or later, you must upgrade the CDA Converter Library from Version 1.1 to Version 2.0.

When installing DECpresent Version 1.0A on OpenVMS VAX Version 6.1 or later, system managers can safely ignore the IVP failure for the CDA Converter Library Version 1.1 because that version of the product is bundled with DECpresent but does not work on OpenVMS VAX Version 6.1 and later.

After installing DECpresent Version 1.0A on OpenVMS VAX Version 6.1 or later, or upgrading from VMS Version 5.5–2 to Version 6.1 or later with DECpresent Version 1.0A already installed on the system, system managers should install CDA Converter Library Version 2.0.

1.3.1.4 VAXstation and MicroVAX Installation Workaround

V6.0 This release note applies specifically to VAXstation 4000-VLC, VAXstation 4000 Model 60, VAXstation 4000 Model 90, MicroVAX 3100 Models 30 and 40, and MicroVAX 3100 Model 80.

If you halt the OpenVMS VAX Version 6.0 (or later) installation procedure after booting the new system disk, the console will either display miscellaneous characters or appear to hang. Turn the system power off and on to recover the use of the console and continue the installation.

Once AUTOGEN runs and the system reboots at the completion of the installation procedure, the console is usable again. ♦

1.4 Installation and Upgrade Information Specific to Alpha

Alpha The release notes in this section pertain only to installations or upgrades of OpenVMS Alpha operating systems. See Section 1.2 for additional notes that pertain to both Alpha and VAX systems. For complete information about installing or upgrading your OpenVMS Alpha Version 7.0 operating system, refer to the *OpenVMS Alpha Version 7.0 Upgrade and Installation Manual*.

1.4.1 Changes and Enhancements

This section describes changes and enhancements to OpenVMS Alpha installation and upgrade procedures.

1.4.1.1 Installing the Instruction Emulator

V7.0 The Instruction Emulator (kit AXPEMUL01_070 on the OpenVMS Alpha Version 7.0 CD–ROM) supplies code that simulates execution of the following Alpha instructions if they are not implemented directly in hardware:

- LDBU and LDWU
- STB and STW
- SEXTB and SEXTW

Some new versions of Alpha compilers can generate these instructions if the code developer specifically requests it. If you run an image with these instructions in it, and your Alpha hardware does not support them, the program will fail with a message similar to the following:

```
%SYSTEM-F-OPCDEC, opcode reserved to Digital fault at PC='location',PS='xxxxxx'
```

The software in kit AXPEMUL01_070 prevents this failure by simulating execution of these instructions. Since instruction emulation is much slower than hardware execution, performance can be poor if you run a program containing these instructions on an Alpha processor that does not support them. The following message displays the first five times that an instruction is emulated:

```
%SYSTEM-I-EMULATED, an instruction not implemented on this processor
was emulated at PC='location', PS='xxxxxx'
```

If you want, you can write a program to collect the following information:

- The total number of emulated instructions executed by an image (saved in the quadword named CTL$GQ_EMULATE_COUNT)
- Addresses of the last 16 emulated instructions (saved in the 16 quadwords starting at CTL$GQ_EMULATE_PC_RING)

  The oldest address is pointed to by CTL$GQ_EMULATE_RING_INDEX.

All of these data are cleared the first time an instruction is emulated in an image.

The AXPEMUL01_070 kit can also be used to remove emulation support and revert to the original OpenVMS Alpha Version 7.0 run-time environment.

Installation Instructions

Kit AXPEMUL01_070 is located in the [EMULATOR] directory of the OpenVMS Alpha Version 7.0 operating system CD–ROM.

Install this kit using the VMSINSTAL utility by logging into the SYSTEM account and entering the following command at the DCL prompt:

@SYS$UPDATE:VMSINSTAL AXPEMUL01_070 *disk*:[EMULATOR]

where *disk* is the name of the CD–ROM drive.

The installation procedure inquires whether you want to apply or remove emulation support. The system should be rebooted after successful installation of the kit. Rebooting any other nodes in your VMScluster enables them to also make use of the new image.

1.4.1.2 Removing the OpenVMS Alpha Operating System

V6.1 Although use of the PRODUCT REMOVE command is not fully supported in OpenVMS Alpha Version 6.1 and later for the removal of the OpenVMS operating system, you can use the PRODUCT REMOVE command to remove most of the OpenVMS Alpha operating system from a system disk without affecting user files on the disk.

Follow these steps to remove OpenVMS Alpha:

1. If your system disk has multiple system-specific roots, boot the system and execute SYS$MANAGER:CLUSTER_CONFIG to remove all roots except the one from which you are booted.

2. Shut down and boot from the distribution CD–ROM or from a system disk other than the one from which OpenVMS Alpha is being removed. Execute the following DCL commands; substitute the device name of the disk from which OpenVMS Alpha is being removed for <target-disk>, and the root number that you did *not* remove in step 1 for SYSx.

   ```
   $ DEFINE/NOLOG PCSI$SYSDEVICE <target-disk>
   $ DEFINE/NOLOG PCSI$SPECIFIC <target-disk>:[SYSx.]
   $ DEFINE/NOLOG PCSI$DESTINATION <target-disk>:[VMS$COMMON]
   $ PRODUCT REMOVE VMS /REMOTE
   ```

   If OpenVMS Alpha is not running from the distribution CD–ROM, you will need to be logged in to a privileged account.

3. After the remove operation completes, review the target disk to determine if you want to delete the following files, which the PRODUCT REMOVE command cannot remove.

   - In <target-disk>:[SYS*.SYSEXE], where * is 0 or the hexadecimal number of any additional VMScluster roots on the target disk:

     ALPHAVMSSYS.PAR
     MODPARAMS.DAT
     PAGEFILE.SYS
     SWAPFILE.SYS

   - In <target-disk>:[VMS$COMMON.SYSEXE]:

     LMF$LICENSE.LDB
     PCSI$FILE_SYSTEM.PCSI$DATABASE
     PCSI$PROCESSOR.PCSI$DATABASE
     PCSI$ROOT.PCSI$DATABASE
     RIGHTSLIST.DAT

   Note

   Do not remove the *.PCSI$DATABASE files if you have layered products installed on this disk, or if you want to maintain a history of software installation on this disk.

4. Review the target disk for the directory structures [VMS$COMMON...] and [SYSx...], which will remain after removing OpenVMS Alpha. You may want to remove these directories.

1.4.2 Problems and Restrictions

This section describes problems and restrictions that apply to installing or upgrading to OpenVMS Alpha Version 7.0.

1.4.2.1 DECram Version 2.2 is Not Supported

V7.0 You cannot install DECram Version 2.2 on OpenVMS Alpha Version 7.0. Version 7.0 of OpenVMS Alpha supports only DECram Version 2.2B.

You also cannot upgrade from OpenVMS Alpha Version 6.2 to Version 7.0 if you have DECram Version 2.2 installed. Attempting to do so prevents the system from booting. A workaround that allows the upgrade to proceed is to use the DCL command RENAME to rename the DECram .EXE file; for example:

```
$ RENAME SYS$SPECIFIC:[SYS$LDR]DECRAM$EXECLET.EXE -
_$ SYS$SPECIFIC:[SYS$LDR]OLD_DECRAM$EXECLET.EXE
```

After renaming the .EXE file, you can install DECram Version 2.2B.

1.4.2.2 RAID Devices Naming Problem

V6.2

If you have RAID devices connected to StorageWorks RAID Array 210 or 230 subsystems, you might experience device-naming problems when running in a cluster environment if nonzero allocation classes are used. In this case, the RAID devices will be named $*n*$DR*cu*, where: *n* is the (nonzero) allocation class, *c* is the controller letter, and *u* is the unit number.

If multiple nodes in the cluster have the same (nonzero) allocation class and these same nodes have RAID controllers, then RAID devices that are distinct might be given the same name (for example, $1$DRA0). This problem can lead to data corruption.

A new system parameter, DR_UNIT_BASE, has been added to provide a way for unique RAID device names to be generated. The RAID controller adds DR_UNIT_BASE to the unit number, *u*, before it creates the RAID device names. With DR_UNIT_BASE set to 0, the device name $1$DRA0 is produced; with the parameter set to 10, the device name $1$DRA10 is produced.

Setting DR_UNIT_BASE to appropriate, nonoverlapping values on all cluster members that share the same (nonzero) allocation class ensures that no two RAID devices are given the same name.

1.4.2.3 Booting Messages

V6.1

You may see the following message on some systems during booting. This is simply an informational message to indicate that an image was not installed as RESIDENT due to insufficient code or data granularity hint region size.

```
%INSTALL-I-NONRES, installed image non-resident with other specified options
-INSTALL-E-NOGHREG, insufficient memory in the code or data granularity hint
region
```

You can either ignore this message or run AUTOGEN to eliminate the error message.

For more information about the /RESIDENT qualifier, see the *OpenVMS System Management Utilities Reference Manual*. For more information on installing resident images, see the *OpenVMS Linker Utility Manual*.

1.4.2.4 Using DECnet on Token Ring Devices

V6.1

DECnet Phase IV is not supported on Token Ring devices. However, to use DECnet over a Token Ring device, you can use the following sample commands:

```
$ MCR NCP
NCP> DEFINE LINE MNA-0 STA ON
NCP> DEFINE CIRCUIT MNA-0 STA ON
NCP> EXIT
$ DEFINE/SYSTEM EXA0 IRA0:
$ @STARTNET
```

The DEFINE/SYSTEM command equates the Token Ring device to the DEMNA Ethernet device, which DECnet Phase IV recognizes. Note that after you enter the DEFINE LINE and DEFINE CIRCUIT commands, you need only include the DEFINE/SYSTEM command prior to the @STARTNET invocation in your system startup command file for the change to take effect across system restarts. ♦

1.5 AlphaServer 2100 Systems (Alpha Only)

Alpha

This section contains release notes that apply to AlphaServer 2100 systems.

1.5.1 Problems and Restrictions

This section describes a hardware restriction pertaining to AlphaServer 2100 systems.

1.5.1.1 SCSI Controller Restriction

V6.2

The Adaptec 1740/1742 SCSI controller (PB2HA–SA) is not supported on AlphaServer 2100 systems having more than 1 gigabyte (GB) of memory. If the controller is connected to such a system, the following message appears on the operator's console:

```
%PKJDRVR-E- The direct DMA window does not map all of memory. Port is going OFF LINE.
```

♦

1.6 DEC 3000 Series Workstations (Alpha Only)

Alpha

This section contains notes pertaining to DEC 3000 Series Workstations.

1.6.1 Changes and Enhancements

The following note describes a change in support.

1.6.1.1 PXG Graphics Board Is No Longer Supported

V7.0

Starting with OpenVMS Version 7.0, the PXG graphics board is no longer supported. ♦

1.7 Ethernet and FDDI Controllers

This section contains release notes that pertain to Ethernet and FDDI controllers.

1.7.1 Problems and Restrictions

This section describes problems and restrictions that pertain to certain Ethernet and FDDI controllers.

1.7.1.1 Multiple Controllers Configured But Not All Attached to Media (Alpha Only)

V6.2

Alpha

If you have multiple Ethernet and FDDI controllers configured on your system, you might experience problems with the InfoServer client transport (LAST) under either of the following conditions:

- Not all of the Ethernet and FDDI controllers are connected to network cabling.
- An FDDI controller is connected to the network cabling, but the FDDI ring is not functional; for example, some FDDI hardware may be powered down or broken.

Problems can range from not being able to access all the services available on the network, if you have four or more controllers configured, to a system crash.

To avoid these problems, specify only the controllers that are attached to media. Digital recommends that you do this by first editing your SYS$STARTUP:ESS$LAST_STARTUP.DAT data file to specify only the controllers that are attached and then restarting your system.

With certain controller configurations, if you specify controllers that are not attached, your system might crash when you issue the following command sequence:

```
$ MC ESS$LASTCP
LASTCP> STOP
```

An example of how to edit the SYS$STARTUP:ESS$LAST_STARTUP.DAT file follows. The unedited file is shown first, followed by an edited file.

```
!++
! This file will be used to set the appropriate LASTCP qualifiers. The following
! LASTCP qualifiers: ALL_CONTROLLERS, CHECKSUM, TRANSMIT_QUOTA, or SLOW_MODE
! can be set by using the following statement format:
! LASTCP qualifier = 1 to enable   e.g. SLOW_MODE = 1 enables  SLOW_MODE
! LASTCP qualifier = 0 to disable  e.g. SLOW_MODE = 0 disables SLOW_MODE
! The remaining LASTCP qualifiers will require the appropriate value settings.
! DEVICE          = (list-of-devices)
! TIMEOUT    = n    minimum interval in seconds
! CIRCUIT_MAXIMUM = n    maximum number of nodes
! GROUP           = n    Group number
! NODE_NAME       = name   Node name
! CONTROLLERS     = ([{controller letter,}...]) Controller list
! TRANSMIT_QUOTA  = n    Number of transmit buffers
!--
ALL_CONTROLLERS = ON
```

The edited SYS$STARTUP:ESS$LAST_STARTUP.DAT file follows. This example assumes you have ESA, ETA, EXA, EZA controllers configured on your system and that only the ESA controller is attached to the Ethernet wire.

```
!++
! This file will be used to set the appropriate LASTCP qualifiers. The following
! LASTCP qualifiers: ALL_CONTROLLERS, CHECKSUM, TRANSMIT_QUOTA, or SLOW_MODE
! can be set by using the following statement format:
! LASTCP qualifier = 1 to enable   e.g. SLOW_MODE = 1 enables  SLOW_MODE
! LASTCP qualifier = 0 to disable  e.g. SLOW_MODE = 0 disables SLOW_MODE
! The remaining LASTCP qualifiers will require the appropriate value settings.
! DEVICE          = (list-of-devices)
! TIMEOUT    = n    minimum interval in seconds
! CIRCUIT_MAXIMUM = n    maximum number of nodes
! GROUP           = n    Group number
! NODE_NAME       = name   Node name
! CONTROLLERS     = ([{controller letter,}...]) Controller list
! TRANSMIT_QUOTA  = n    Number of transmit buffers
!--
ALL_CONTROLLERS = OFF
DEVICE = (ESA)
```

Note

The default ESS$LAST_STARTUP.DAT file is stored in SYS$COMMON:[SYS$STARTUP]. You might want to put the edited file in SYS$SPECIFIC:[SYS$STARTUP]. Otherwise, other system roots might be affected. ♦

1.8 RF73 and Other RF*nn* DSSI Disk Devices

Notes in this section pertain to the RF31T, RF31T+, RF35, RF35+, RF73, and RF74 DSSI disk devices.

1.8.1 Problems and Restrictions

This section describes a problem found in certain RF31T, RF31T+, RF35, RF35+, RF73, and RF74 DSSI disk devices.

1.8.1.1 RF73 and Other RF*nn* DSSI Disk Devices and Controller Memory Errors

V6.2 A problem exists with the microcode for earlier versions of RF31T, RF31T+, RF35, RF35+, RF73, and RF74 DSSI disk devices that can cause data loss. The problem can occur when reading data from one of these devices when the device has had a controller memory error (also known as an error detection and correction (EDC) error). The error could have been induced by a virtual circuit closure or faulty hardware.

Digital advises customers with any of these devices to check their microcode revision levels. If the microcode revision levels are lower than the numbers shown in Table 1–1, Digital recommends that you update the microcode. The microcode for all models, except the RF31T, RF31T+, and RF35+, is provided on the latest OpenVMS binary distribution CD–ROM.

The RF_VERS utility, a utility program that displays the microcode revision level of the DSSI disk devices, is also provided on the CD–ROM. Instructions for using the utility program and for updating the microcode are provided in this section.

Note

If you have an RF31T, RF31T+, or RF35+ disk drive with a version of microcode that is not error-free (see Table 1–1), and if you have a support contract, contact your Digital support representative. Otherwise, contact your Digital account representative or your authorized reseller.

The earliest supportable revision levels of the DSSI disk microcode are shown in Table 1–1.

Table 1–1 Error-Free Microcode Revision Levels

Device Type	Minimum Level with Error-Free Microcode
RF31T	T387E
RF31T+	T387E
RF35	T392D
RF35+	T392D
RF36	V427P
RF73	T392D
RF74	V427P

To display the microcode version level of your DSSI disk devices, perform the following steps:

1. Log in to the SYSTEM account or another account that has the CMKRNL, DIAGNOSE, and SYSPRV privileges.

2. Issue the following commands:

```
$ SET PROCESS /PRIVILEGE=(DIAGNOSE,CMKRNL,SYSPRV)
$ SHOW DEVICE FYA0:
```

VAX

On VAX systems, if the SHOW DEVICE command produces an error, issue the following commands:

```
$ RUN SYS$SYSTEM:SYSGEN
        SYSGEN>  CONN FYA0/NOADAP
        SYSGEN>  ^Z ♦
```

Alpha

On Alpha systems, if the SHOW DEVICE command produces an error, issue the following commands:

```
$ RUN SYS$SYSTEM:SYSMAN
        SYSMAN> IO CONNECT FYA0: /NOADAP
        SYSGEN>  ^Z ♦
```

3. On VAX and Alpha systems, issue the following command:

```
$ RUN SYS$ETC:RF_VERS.EXE
```

The following is an example of the display produced by the RF_VERS utility:

```
Program Name:   RF_VERS
Revision Level: V1.2s

NOTICE: This program does not currently support the RF72 or any
        HSDxx controllers. See next version for support.

DSSI disks currently on this system as seen by RF_VERS

Device           Node        Status           Hardware    Firmware
Name             Name                         Type        Version

_$22$DIA7:       R4JL2I      mounted          RF73        T387A
_$22$DIA6:       R4I0BG      mounted          RF73        T387A
_$22$DIA8:       R4XLWE      mounted          RF73        T387A
_$22$DIA2:       R4FCZK      mounted          RF73        T387A
_$22$DIA3:       R4CKCG      mounted          RF73        T387A
_$22$DIA4:       R4ZKUE      mounted          RF73        T387A
_$22$DIA9:       R4GYYI      mounted          RF73        T387A
_$22$DIA1:       R4XRYI      mounted          RF73        T387A
```

To update the microcode in your device, use the appropriate command for your device and platform from Table 1–2.

Caution

Back up the disk before updating the microcode.

Table 1–2 Commands for Updating Microcode in Certain DSSI Disk Devices

Device Type	Platform	Command
RF35	Alpha	$RUN SYS$ETC:RF35_T392F_DEC_ALPHA.EXE
RF35	VAX	$RUN SYS$ETC:RF35_T392F_DEC.EXE
RF36	Alpha	$RUN SYS$ETC:RF36_V427P_DEC_ALPHA.EXE
RF36	VAX	$RUN SYS$ETC:RF36_V427P_DEC.EXE
RF73	Alpha	$RUN SYS$ETC:RF73_T392F_DEC_ALPHA.EXE
RF73	VAX	$RUN SYS$ETC:RF73_T392F_DEC.EXE
RF74	Alpha	$RUN SYS$ETC:RF74_V427P_DEC_ALPHA.EXE
RF74	VAX	$RUN SYS$ETC:RF74_V427P_DEC.EXE

Caution

Do not delete SCSI_INFO.EXE, RF_VERS.EXE, or any of the files listed in Table 1–2. If these files are deleted, VMSKITBLD.COM (on VAX) will not be able to find them. Similarly, on Alpha systems, the PRODUCT INSTALL commands in AXPVMS$PCSI_INSTALL and AXPVMS$PCSI_INSTALL_MIN will fail.

1.9 LAT Support for DECservers

This section contains notes pertaining to LAT support for DECservers.

1.9.1 Changes and Enhancements

The following note describes a change in LAT baud rate support.

1.9.1.1 Increased Baud Rate Support on LAT

V7.0 LAT Version 5.3 contains increased support for baud rates up to 4,294,967,295. Previously LAT could report speeds only up to 57600. As soon as corresponding support is added to terminal servers, OpenVMS users can take advantage of this increased support.

2

General User Release Notes

This chapter provides information for all users of the OpenVMS operating system. It includes information about commonly used commands and utilities.

For information about new features included in this version of the software, refer to *OpenVMS Version 7.0 New Features Manual*.

2.1 DCL Commands

This section contains release notes related to the DIGITAL Command Language (DCL) for this release of the OpenVMS operating system.

2.1.1 Problems and Restrictions

The notes in this section describe problems and restrictions pertaining to DCL commands.

2.1.1.1 SET PROCESS/NOAUTO_UNSHELVE Command in Cluster Environment

V6.1

The command SET PROCESS/NOAUTO_UNSHELVE as implemented in Version 6.1 does not support operations across the cluster. It can be issued only for a process on the same node, including as the default case, the process from which the command is issued.

The /IDENTIFICATION=pid qualifier is supported, but only when the target process is on the same node as the process where the command is issued.

2.1.1.2 SET PROCESS/SUSPEND=KERNEL/ID= Command in Cluster Environment

VAX V6.0
Alpha V1.5

When you issue the SET PROCESS/SUSPEND=KERNEL/ID= command in a cluster environment, the KERNEL keyword is ignored if the target process and the current process reside on different cluster nodes. As a result, process suspension is handled as if you had specified the SUPERVISOR keyword (the default).

This is caused by a problem with the $SUSPND system service, as discussed in Section 4.28.1.2. Digital expects to fix this problem in a future version of the OpenVMS operating system.

2.2 DECforms

This section contains notes pertaining to DECforms.

2.2.1 Problems and Restrictions

The following note describes a DECforms support issue.

2.2.1.1 DECforms Support on OpenVMS Version 7.0 (Alpha Only)

V7.0

Alpha

Because of changes in DECthreads, DECforms Version 2.1 does not work with OpenVMS Alpha Version 7.0. Installing OpenVMS Alpha Version 7.0 causes existing applications based on DECforms Version 2.1 to fail; installing DECforms Version 2.1 on OpenVMS Alpha Version 7.0 also fails. In both cases, you get the following error message:

```
%CMA-F-USE_ERROR, requested operation is inappropriate for the specified object
```

In order for DECforms-based applications to operate correctly on OpenVMS Alpha Version 7.0, you must run DECforms Version 2.1A.

DECforms Version 2.1A currently ships as a supplemental CD–ROM in the OpenVMS Alpha Software Library Package. The supplemental CD–ROM has also been distributed to customers who purchased OpenVMS software library update services. DECforms Version 2.1A will be included in the regular OpenVMS Software Library Package scheduled for March 1996. Please contact your Digital support representative for more information. ♦

2.3 DECnet Layered Product

This section contains release notes pertaining to the DECnet layered product.

2.3.1 Changes and Enhancements

The notes in this section describe changes in support for the DECnet layered product.

2.3.1.1 DECnet Phase IV Superseded by DECnet/OSI

V6.2

The DECnet/OSI layered product supersedes the DECnet Phase IV product distributed with the OpenVMS operating system. DECnet Phase IV still ships with the OpenVMS operating system, but it will be retired in a future release.

DECnet/OSI (DECnet Phase V) was first released on OpenVMS in 1991. As with all previous releases of DECnet products, there is full backwards compatibility with the previous version. As such, customers can continue to use Phase IV network protocols and applications with DECnet/OSI.

The current release of DECnet/OSI provides a suite of tools for ease of upgrade and ease of use as customers make the transition to the new environment. Future releases will incorporate additional enhancements and improvements based on customer feedback and needs.

DECnet/OSI for OpenVMS on VAX and Alpha platforms provides the functionality of DECnet plus support of the international OSI protocols. For more information about upgrading to DECnet/OSI, see Section 1.2.2.1.

2.4 DECTPU

This section contains notes pertaining to the DEC Text Processing Utility (DECTPU).

2.4.1 Problems and Restrictions

The notes in this section describe DECTPU problems and restrictions.

2.4.1.1 Motif Widget Context Help Built-In

V1.0

The following built-in, which should enter Motif context-sensitive help mode, is disabled because of a problem in the Motif toolkit:

```
SET (WIDGET_CONTEXT_HELP, widget_variable, {on|1|off|0})
```

The mouse pointer changes to a question mark, and DECTPU waits for you to select a widget by clicking MB1. DECTPU then executes the help callback of the selected widget (or of its parent if the selected widget has no help callback). The widget_variable is the widget within which the modal help interaction will occur, usually the top-level widget returned from the GET_INFO (SCREEN, “widget”) built-in. The last parameter confines the question mark pointer to the specified widget if ON or 1, and does not confine the pointer if OFF or 0.

2.5 DECwindows Motif for OpenVMS

This section contains several notes of general interest pertaining to the DECwindows Motif for OpenVMS layered product. All other notes for this product are included in Section 3.16 and related sections.

2.5.1 Changes and Enhancements

This section includes notes about support for the DECwindows Motif for OpenVMS layered product.

2.5.1.1 DECwindows Motif Version 1.1 for OpenVMS No Longer Supported

V7.0

Starting with OpenVMS Version 7.0, the OpenVMS operating system no longer supports DECwindows Motif Version 1.1 for OpenVMS. To use DECwindows with OpenVMS Version 7.0 and later, you must install DECwindows Motif Version 1.2 or later. However, DECwindows Motif Version 1.2 and later provide run-time support for programs built on earlier versions of DECwindows and DECwindows Motif. For more information, see the *DECwindows Motif for OpenVMS Release Notes* for Version 1.2 or later. See Section 3.16.2.1 for a Version 1.2 restriction.

2.5.1.2 NCSA Mosaic Browser: TCP/IP Support Options

V7.0

DECwindows Motif Version 1.2–3 includes the NCSA Mosaic Version 2.4 web browser. The *DECwindows Motif Version 1.2–3 for OpenVMS Release Notes* state that the NCSA Mosaic browser requires that the TCP/IP Services for OpenVMS product (UCX) be installed on your system. Note that TCP/IP support can alternatively be provided by PathWay for OpenVMS (Wollongong), TCPware (Process Software), or MultiNet Version 3.4B (TGV).

2.5.2 Problems and Restrictions

This section describes a problem with using DECwindows to access Notes.

2.5.2.1 DEC Notes Fails With LinkWorks Error Message (Alpha Only)

V7.0

Alpha

If you run DECwindows Motif Version 1.2–3 for OpenVMS on an OpenVMS Alpha Version 7.0 system and use the DECwindows interface to DEC Notes, Notes fails to open remote notes conferences and displays the following error message:

```
 LinkWorks Reported Error: Unknown error
```

To correct this problem, install patch kit AXPMOTF06_U3012, which is on the OpenVMS Version 7.0 CD–ROM in the same directory as the DECwindows Motif V1.2-3 for OpenVMS Alpha kit. ♦

2.6 High-Performance Sort/Merge Utility (Alpha Only)

Alpha

This section contains release notes pertaining to the command line interface of the OpenVMS Alpha high-performance Sort/Merge utility. See the *OpenVMS Version 7.0 New Features Manual* for information about using this utility.

See Section 4.13.1 for release notes applicable to the SOR$ callable interface to the high-performance Sort/Merge utility.

2.6.1 Problems and Restrictions

This section describes problems associated with the command line interface to the OpenVMS Alpha Version 7.0 release of the high-performance Sort/Merge utility.

2.6.1.1 Default File Specification

V7.0

Unlike the Sort/Merge utility, the high-performance Sort/Merge utility does not take its default output file extension from the first input file (in instances where there is a user-specified input file). Therefore, you should explicitly specify the output file extension.

This difference in behavior will be fixed in a future major release.

2.6.1.2 Error Messages

V7.0

The following problem pertains to error messages generated by the high-performance Sort/Merge utility. Error messages that would normally refer to one or more files by name, instead contain the FAO directive that would be used to generate the file name. In addition, error messages do not include RMS or other secondary error messages. For example, if the high-performance Sort/Merge utility cannot find the X.DAT input file, it generates the message:

```
%SORT-F-OPENIN, error opening !AS as input
```

The Sort/Merge utility generates a message that contains the file name:

```
%SORT-F-OPENIN, error opening DISK$2:[SORT_AREA]X.DAT; as input
-RMS-E-FNF, file not found
```

This difference in behavior will be fixed in a future major release.

2.6.1.3 Concurrent Sort Operations

V7.0

Memory allocation differences may limit the high-performance Sort/Merge utility's ability to perform the same number of concurrent sort operations as can be performed by the Sort/Merge utility in the same amount of virtual memory.

If this situation occurs, you can either increase the amount of virtual memory that is available to the process, or reduce the working set extent (WS_EXTENT).

2.6.1.4 Merging Stream Files Limitation

V7.0

A limitation exists for merging stream files in this release of the high-performance Sort/Merge utility. The end of file is not written correctly for the output stream file. To work around this limitation, the format of the output file should be explicitly specified as fixed or variable. This can be done at the command line by using one of the following commands:

```
   $ merge ... out_file/format=fixed:n
   $ merge ... out_file/format=variable:n
```

If you want the output file in stream format, the RMS Convert utility can be used to convert the output file from fixed or variable record format to stream format.

This limitation will be removed in a future major release. ♦

2.7 Layered Product Support

V7.0

Information about layered product support is now available in the Software Public Rollout Reports for OpenVMS, available on the World Wide Web. The Software Public Rollout Reports for OpenVMS list the availability of Digital's software products shipping on the Software Products Library kits (CD–ROM consolidations) for OpenVMS Alpha and OpenVMS VAX.

Tables in these reports show layered product support for operating system versions, including OpenVMS Version 7.0. The reports contain the product name and version, the operating system version required to support the product, and the volume ship date for the product. The information in these tables is continually evolving and is subject to change. The reports are intended for public distribution and are updated monthly.

These reports are available from the OpenVMS World Wide Web in the "literature" section. Use the following URL:

```
http://www.openvms.digital.com
```

If you do not have Internet access, you can find the operating system support information on any of the quarterly Software Products Libraries, in the following directory:

```
[README]SW_COMPAT_MATRIX.PS (.TXT)
```

The Software Public Rollout Reports are also available from your Digital support representative.

2.8 Mail Utility

This section contains release notes pertaining to the OpenVMS Mail utility. Notes of particular interest to system managers are documented in Section 3.22.

2.8.1 Changes and Enhancements

This section describes changes and enhancements to Mail.

2.8.1.1 Specifying Transports

V7.0

Beginning with OpenVMS Version 6.2, Mail would interpret a specified address as follows:

- If the node component of the address contained a period (.), the address would be interpreted as an Internet address. Mail would then use the SMTP protocol by default unless you had previously set up your system to use a different Internet protocol by defining that alternate protocol with the MAIL$INTERNET_TRANSPORT logical name.
- If the node component of the address did not contain a period, the address would be interpreted as a DECnet address.

With the release of OpenVMS Version 7.0, the default behavior is still in effect (that is, the system chooses the Internet or DECnet protocol based on the presence or absence of a period), but you can now force the system to choose a specific protocol. This option is helpful in cases where a mail address can be interpreted as valid for either the Internet or DECnet protocol.

To specify protocols, you can define the MAIL$INTERNET_MODE logical name as follows:

- HYBRID (the default)

  If the node component of the address contains a period (.), Mail uses an Internet protocol. If there are no periods, Mail uses the DECnet protocol.

- DECNET

  Mail always interprets the node component of the address as a DECnet node specification.

- SMTP

  Mail always interprets the node component of the address as an Internet address specification. The default transport is SMTP unless you use the MAIL$INTERNET_TRANSPORT logical to define an alternate Internet transport.

To modify your Mail environment in any of these ways, Digital recommends that you define the MAIL$INTERNET_MODE and MAIL$INTERNET_TRANSPORT logicals in your LOGIN.COM file. For example, if your system is set up to use the default (HYBRID), the Mail address smith@pluto is interpreted as a DECnet address because there are no periods in that address. However, if you want Mail to use SMTP instead of DECnet, you can add the following line to your LOGIN.COM file:

```
$ DEFINE MAIL$INTERNET_MODE SMTP
```

When you then specify smith@pluto, Mail interprets this address as an Internet address and uses the SMTP protocol (for example, SMTP%"smith@pluto.xyz.dec.com").

2.8.1.2 SET FORWARD Command: Specifying Quotation Marks

V7.0 In prior versions of the OpenVMS operating system, you had to specify an extra pair of quotation marks if you wanted them included with the SET FORWARD command because the command automatically removed the first pair. Starting with OpenVMS Version 7.0, you need not specify an extra pair of quotation marks because the SET FORWARD command no longer removes the first pair.

2.8.2 Problems and Restrictions

The following note describes how to fix a problem in Mail.

2.8.2.1 Replying to an Address Containing Nested Quotation Marks

V7.0 In most cases, you can use the Mail command REPLY to reply to mail received from an address containing nested quotation marks. However, if your system does not have this capability, contact your system manager.

The system manager can correct the problem by setting bit 5 (bitmask value 16) in the logical MAIL$SYSTEM_FLAGS (shown in the example in SYS$MANAGER:SYLOGICALS.TEMPLATE).

3

System Management Release Notes

This chapter contains information that applies to system maintenance and management, performance management, and networking.

For information about new features included in this version of the software, refer to the *OpenVMS Version 7.0 New Features Manual*.

3.1 Audit Analysis Utility

This section contains release notes pertaining to the Audit Analysis utility (ANALYZE/AUDIT).

3.1.1 Documentation Changes and Corrections

This section describes corrections and additions to the Audit Analysis utility documentation.

3.1.1.1 *OpenVMS System Management Utilities Reference Manual: A–L*

V7.0 The following additions and corrections to the *OpenVMS System Management Utilities Reference Manual: A–L* pertain to subtypes used by the Audit Analysis utility:

- Insert the following description of the SUBTYPE criterion under the description of the ANALYZE/AUDIT command's /SELECT qualifier. Note that the SUBTYPE criterion can also be used with the /IGNORE qualifier.

  SUBTYPE=(subtype,...)
  Specifies the type of audit records you wish to include or exclude (if used with the /IGNORE qualifier). The subtypes are listed in Table F-2.

  For example, the following command selects batch and detached processes for display:

  ```
  $ ANALYZE/AUDIT/SELECT=SUBTYPE=(BATCH,DETACHED)
  ```

  The next command excludes dialup and remote processes:

  ```
  $ ANALYZE/AUDIT/IGNORE=SUBTYPE=(DIALUP,REMOTE)
  ```

- In Table F-2, correct the subtype names by deleting the NSA$C_ prefix from all of them. When specifying these subtypes with the new SUBTYPE criterion described earlier in this note, you must enter the subtypes *without* the NSA$C_ prefix (as demonstrated in the preceding examples).

3.2 Authorize Utility

This section contains release notes pertaining to the Authorize utility (AUTHORIZE).

3.2.1 Changes and Enhancements

The following note describes a change in the Authorize utility.

3.2.1.1 PQL_MWSEXTENT Can Override WSEXTENT

V7.0 AUTOGEN now sets the value of the SYSGEN parameter PQL_MWSEXTENT to be equal to the value of WSMAX unless you override PQL_MWSEXTENT by settings in MODPARAMS.DAT.

When a user logs in, the system creates the interactive process and sets its working set extent by checking both the PQL_MWSEXTENT value and the user account's value of WSEXTENT and choosing the *larger* value.

Increasing the value of PQL_MWSEXTENT allows all subsequent processes created on a system to allocate more physical memory before paging begins. As a result, you might see processes using more physical memory than was defined for them by the settings in AUTHORIZE. Allocating more memory often improves system performance and perceived system responsiveness, but some system managers might prefer to use other values for this parameter.

Because the parameter PQL_MWSEXTENT is dynamic, you can change its value without reloading the system. However, you should also add a definition to MODPARAMS.DAT to ensure that the parameter retains its desired value after running AUTOGEN and after system reboots.

3.3 AUTOGEN Command Procedure

This section contains release notes pertaining to the AUTOGEN command procedure.

3.3.1 Changes and Enhancements

This section describes changes and enhancements to AUTOGEN on OpenVMS systems.

3.3.1.1 AGEN$PARAMS.REPORT: Turning Off Logging of DCL Statements

V7.0 The contents of MODPARAMS.DAT are evaluated as DCL statements, and you can make assignments to symbols with names that are not SYSGEN parameters (for example, scratch variables or conditional assignments based on other values). Traditionally, every such assignment is logged in AGEN$PARAMS.REPORT, sometimes creating a large file with many logging statements that do not interest users.

Starting with OpenVMS Alpha Version 6.2 and OpenVMS VAX Version 7.0, you can designate any assignments that you prefer not to log in AGEN$PARAMS.REPORT by prefixing every such assignment with a dollar sign ($). When AUTOGEN encounters a MODPARAMS.DAT record beginning with a $, it does not check the list of known SYSGEN parameters and does not log this record to AGEN$PARAMS.REPORT.

3.3.1.2 Computation of PQL_DPGFLQUOTA and PQL_MPGFLQUOTA (VAX Only)

V7.0

VAX

AUTOGEN now sets the value of PQL_DPGFLQUOTA to be the larger of either its default value or the value computed for PQL_DWSEXTENT, and sets the value of PQL_MPGFLQUOTA to be the larger of either its default value or the value computed for PQL_MWSEXTENT. ♦

3.3.1.3 NPAGEDYN and NPAGEVIR Limitations and Warnings (VAX Only)

V7.0

VAX

For the benefit of systems with limited physical memory, AUTOGEN logs a warning message in its report if NPAGEDYN exceeds 10% of physical memory or if NPAGEVIR exceeds 33% of physical memory.

AUTOGEN also limits its own calculated value for NPAGEDYN to 20% of physical memory, and limits NPAGEVIR to 50% of physical memory. These calculated values are adequate for most workstations and systems with 16 or fewer megabytes of physical memory. If your system requires a larger value, you can override the AUTOGEN calculated values by setting higher values in MODPARAMS.DAT. ♦

3.3.1.4 Calculating the Page File Size (Alpha Only)

V7.0

Alpha

On Alpha systems, the formula for calculating the size of page file space has changed. The new formula is as follows:

```
size-in-blocks (total for all page files on the system)
= physical-memory-size (in pagelets)
+ 8192 (supplementary amount)
```

To calculate the physical memory size in pagelets, follow these steps:

1. Enter the following command:

   ```
   $ SHOW MEMORY/PHYSICAL_PAGES
   ```

   The number of physical pages is listed in the Total column.

2. To compute the number of pagelets per page, divide the system page size by 512 (pagelet size). For example, a system with a page size of 8192 has 16 pagelets per page.

   To determine a system's page size, enter the following command:

   ```
   $ WRITE SYS$OUTPUT F$GETSYI ("PAGE_SIZE")
   ```

3. Multiply the number of pagelets per page by the number of physical pages. (The physical page value is in the Total column in the SHOW MEMORY /PHYSICAL_PAGES display.)

Adding 8192 to the physical memory size provides an extra margin of safety during periods of heavy paging activity.

After making the initial calculation, observe your system over time and make adjustments as necessary. For more information about calculating the size of page file space, refer to the *OpenVMS System Manager's Manual: Tuning, Monitoring, and Complex Systems*. ♦

3.3.1.5 Paging File Size Computation (VAX Only)

V7.0

VAX

AUTOGEN computes the total size of paging files based on both the size of VIRTUALPAGECNT and the previous usage of installed paging files stored as part of the feedback data.

Starting with OpenVMS Version 7.0, AUTOGEN uses a new algorithm to compute VIRTUALPAGECNT. On machines with larger amounts of memory, this new algorithm computes a smaller VIRTUALPAGECNT value than did previous versions of OpenVMS. AUTOGEN computes a smaller size for paging files if the feedback information indicates that the paging files had not been fully used (or if there is no feedback information).

Because AUTOGEN cannot remove a portion of the paging file, AUTOGEN creates a new file, and leaves the old file to be purged after a reload.

The new algorithm is most useful for machines using one of the smaller disks as a system disk. Depending on your specific configuration, you might prefer to continue using the paging files sized using the older algorithm. If you choose to use paging files sized using the older algorithm, you must define PAGEFILE or MIN_PAGEFILE appropriately in MODPARAMS.DAT before beginning the OpenVMS upgrade. ♦

3.3.1.6 VIRTUALPAGECNT Computations (VAX Only)

V7.0

VAX

The computation of VIRTUALPAGECNT is no longer linear but resembles a logarithmic curve. Instead of computing VIRTUALPAGECNT as being equal to the size of physical memory, it is now computed as the sum of the first 32 MB, plus a quarter of the memory from 32 to 256 MB, plus a sixteenth of the memory (if any) above 256 MB.

This is intended to assist managers of systems that host large numbers of users whose working sets are not large. Systems whose user bases consist of a small number of users (or processes) that require large amounts of physical memory (for example, simulations) might need to set MIN_VIRTUALPAGECNT to a value that satisfies the requirements of those processes. ♦

3.3.1.7 WSMAX Computations

V7.0

Starting with OpenVMS Alpha Version 7.0 and OpenVMS VAX Version 6.2, the computation of WSMAX is no longer linear but resembles a logarithmic curve. Instead of computing WSMAX as being a quarter of the size of physical memory, it is now computed as a quarter of the first 32 MB, plus a sixteenth of the memory from 32 to 256 MB, plus a sixty-fourth of the memory (if any) above 256 MB.

This is intended to assist managers of systems that host large numbers of users whose working sets are not large. Systems whose user bases consist of a small number of users (or processes) that require large amounts of physical memory (for example, simulations) might need to set MIN_WSMAX to a value that satisfies the requirements of those processes.

3.3.2 Problems and Restrictions

This section describes known AUTOGEN problems and restrictions.

3.3.2.1 Full Memory Dumps (Alpha Only)

V6.2

Alpha

Starting with Version 6.2, OpenVMS Alpha does not perform full memory dumps (the system parameter DUMPSTYLE set to 0 or 2) on systems with over 4 gigabytes of memory. Selective dumps (the system parameter DUMPSTYLE set to 1 or 3) are supported for all memory sizes.

The default action by AUTOGEN is to set DUMPSTYLE to 1. This is the recommended setting for all systems. ♦

3.4 Backup Utility

Notes in this section pertain to the Backup utility (BACKUP).

3.4.1 Problems and Restrictions

This section describes known problems and restrictions for the Backup utility.

3.4.1.1 Image and Incremental Backups

V6.2

The first time you back up a disk, you must perform an image backup using the BACKUP/IMAGE/RECORD command before you perform regular incremental backups. The image backup saves a copy of the entire disk and marks each file as being saved. Subsequent incremental backups assume an image backup has been performed and, therefore, save only new or modified files.

If an image backup was not performed first, the incremental backups save more files than might be necessary to ensure that an incremental restore will be successful.

3.4.1.2 Incremental Backups Using PATHWORKS for OpenVMS Servers

V6.2

An incompatibility between the operating procedures of the PATHWORKS for OpenVMS Macintosh server and OpenVMS incremental backup operations can cause BACKUP to save entire disks or directory structures, including subdirectories and files.

A recent change to fix other problems now causes Backup to detect whether a directory file has been modified since the date indicated by the Backup Date field in the file header. If a directory file has been modified, all subdirectories and files of that directory are saved for possible later restore operations. Updating the modification date of directory files is unusual for OpenVMS systems, but it can happen, for example, if you rename a directory file from one location to another. By contrast, the PATHWORKS Macintosh server maintains the modification date of directory files for Macintosh users; that is, it updates the modification date for each directory change, file creation, and file deletion.

Thus, an incremental backup of a disk where PATHWORKS is used to serve files to Macintosh users may result in saving the entire disk or entire directories (including their subdirectories and files) instead of just the user files that were created or modified since the last incremental backup operation.

This incompatibility will be addressed in a future version of OpenVMS.

For now, you can avoid needless saving of files by performing a dummy BACKUP /RECORD operation on all directory files *immediately* before performing the incremental backup. The following example illustrates this workaround:

```
$ BACKUP/RECORD/IGNORE=(INTERLOCK) -
_$ disk:[000000...]*.DIR;* -
_$ NLA0:DUMMY.BCK/SAVE/NOCRC/GROUP_SIZE=0
$
$ BACKUP/VERIFY/FAST/RECORD/IGNORE=(INTERLOCK) -
_$ /NOASSIST/COMMENT="Incremental backup of DISK:" -
_$ disk:[000000...]*.*;*/SINCE=BACKUP -
_$ tape:incr.bck/LABEL=incr/SAVE
```

In this example, the first BACKUP command performs the dummy backup operation and the second performs the actual incremental backup. The first command updates the Backup Date field for all the directory files. Specifying the null output device [000000...] causes no saveset file to actually be written. Since no file information need be retained from this operation, the /NOCRC and /GROUP_SIZE=0 qualifiers are specified to avoid CRC and XOR block calculation.

3.4.1.3 Warning Issued on ANALYZE/DISK Operation

V6.2

An ANALYZE/DISK operation performed immediately after a BACKUP/IMAGE restore of a disk might result in a warning message similar to the following:

```
%ANALDISK-W-ALLOCCLR, blocks incorrectly marked allocated
       LBN 97 to 105, RVN 1
```

This can be caused by attempting to perform a BACKUP/IMAGE restore operation where alias file entries are restored as separate (primary) file entries. (The primary file, which uses the same file header but allocates different data storage blocks, is also restored.)

Note that there is no BACKUP error or loss of data.

This problem will be addressed in a future version of OpenVMS.

3.4.1.4 CD–ROM Menu System (VAX Only)

V6.1

VAX

Do not use the CD–ROM menu system, which displays when you boot the OpenVMS VAX operating system CD–ROM, to back up user disks. Use it to back up *system* disks only.

When you boot from the SYS1 directory on the distribution CD–ROM, you are booting a writelocked system disk that does not allow paging. Because of this, the system displays error messages similar to the following:

```
%SYSINIT-E, error opening page file, status = 0000025C
%SYSINIT-E, error opening swap file, status = 0000025C
%SYSINIT, primary PAGEFILE.SYS not found; system initialization continuing
%SYSINIT, no dump file - error log buffers not saved
%SYSINIT-E, error mounting system device, status = 00000F64
```

These messages are normal. The lack of paging and swap files does not affect most operations.

If you back up large user disks, BACKUP may need to page and the operation could fail. Use online BACKUP to back up user disks.

This can also occur when you use the CD–ROM menu system to back up large system disks on low memory systems (those with less than 32 MB of memory). If this problem occurs, use standalone BACKUP to back up system disks. ♦

3.4.1.5 Image Backups from an RF73 Disk

V6.1

When performing an image backup from an RF73 disk (or a disk with a cluster size of 4 blocks) to an RF74 disk (or a disk with a cluster size of 7 blocks), the Backup utility does not check the file size when it is allocating space for the file being copied. Therefore, if the file has an allocation greater than the value of the CLUSTER_SIZE attribute established during initialization, the Backup utility will allocate one more cluster size number of blocks to the allocation size even though the actual file size is less than the cluster size. For example, during an image backup, a file that uses 6 blocks and is allocated 8 blocks (which displays as 6/8 on the screen if you enter a DIRECTORY/SIZE=ALL command) shows an increase in its allocation size to 14, instead of 7, after it is copied to the target disk.

As a result of this problem, the following files are copied to the image system disk with a blocks used/allocation size of 6/14 blocks:

SYS$COMMON:[SYS$LDR]LIDRIVER.EXE
SYS$COMMON:[SYS$LDR]LPDRIVER.EXE

This incorrect allocation size causes standalone BACKUP to *fail* on the booted image system disk.

To correct this problem, recopy the two previously listed files to the same directory after the image backup, by using the following command (which also specifies the correct allocation size):

```
$ COPY/ALLOCATION=7 SYS$COMMON:[SYS$LDR]LIDRIVER.EXE SYS$COMMON:[SYS$LDR]
$ COPY/ALLOCATION=7 SYS$COMMON:[SYS$LDR]LPDRIVER.EXE SYS$COMMON:[SYS$LDR]
```

3.4.1.6 Relative File Version -0 Processed Like 0

V6.1

The Backup utility processes relative version -0 as if it were 0, saving the most recent version instead of the earliest version of the file for processing.

3.4.1.7 Standalone BACKUP Version 5.3 on Console Media (VAX Only)

V6.1

VAX

OpenVMS VAX Version 6.1 and later is available on 9-track magtape in three forms: magnetic tape only or with standalone BACKUP on either RX50 or RL02 console media. The standalone BACKUP on the console media is VMS Version 5.3; the media is so labeled and identifies itself as Version 5.3 on startup. The standalone BACKUP console media have not been updated for OpenVMS VAX Version 6.1 and later.

OpenVMS VAX Version 6.1 and later is not available with standalone BACKUP on RX01 or TU58 console media. Installations using these console media are expected to have standalone BACKUP console media created from previous versions of the operating system. If you need standalone BACKUP on RX01 or TU58 media and are not in a position to create your own kits before installing Version 6.1 or later, please contact your Digital support representative for assistance.

After you have installed OpenVMS VAX Version 6.1 or later, you can build a Version 6.1 or later standalone BACKUP kit on a disk of your choice or on the RL02 or RX50 console media. You cannot build a Version 6.1 or later standalone BACKUP kit on RX01 or TU58 media. If you use either of the latter, you must retain a kit built from a previously installed version of the operating system. ♦

3.4.1.8 VMS$COMMON.DIR File: Restore Problems

VAX V6.0
Alpha V6.1

On an OpenVMS system disk, the file [SYSx]SYSCOMMON.DIR is an alias directory of the file [000000]VMS$COMMON.DIR. This means that both directory entries point to the same file header. Prior to OpenVMS VAX Version 5.5-2 and OpenVMS Alpha Version 1.5, BACKUP did not properly restore the VMS$COMMON.DIR file. Although this does not affect the system disk, it might produce errors with DIGITAL Command Language (DCL) lexical functions.

OpenVMS VAX Version 5.5-2 and OpenVMS Alpha Version 1.5 corrected this problem. However, if you restore image backups created with an old OpenVMS version, the problem can recur.

The symptoms of the problem are different depending on which version of the operating system you are using. If you upgraded from OpenVMS VAX Version 5.5-2 or OpenVMS Alpha Version 1.5 to OpenVMS VAX Version 6.0 or later or OpenVMS Alpha Version 6.1 or later, it is unlikely that your system disk has this problem. However, you should confirm this and correct the problem if necessary.

To restore VMS$COMMON to its proper state, enter the following commands:

```
$ SET DEFAULT DISK:[000000]
$ SET FILE/ENTER=SYSCOMMON.DIR VMS$COMMON.DIR
$ SET FILE/REMOVE VMS$COMMON.DIR;
$ RENAME SYSCOMMON.DIR VMS$COMMON.DIR
```

If you upgraded to OpenVMS Version 6.1 or later from OpenVMS VAX Version 5.5-2 or OpenVMS Alpha Version 1.5 without following this procedure, your system disk could be affected by this problem.

To determine if your system disk has this problem, enter a BACKUP/LIST command to display save set information about the files contained in the VMS$COMMON directory. For example:

```
   .
   .
   .
[000000]VOLSET.SYS;1                                         0  24-SEP-1994 19:31
[]000000.DIR;1                                               1  24-SEP-1994 19:31
[]SYSCOMMON.DIR;1                                            2  24-SEP-1994 19:31
[]SYSLIB.DIR;1                                              18  24-SEP-1994 19:31
[]SYSTEST.DIR;1                                              1  24-SEP-1994 19:31
[]SYSMAINT.DIR;1                                             1  24-SEP-1994 19:31
[]SYSMGR.DIR;1                                               6  24-SEP-1994 19:31
[]SYSHLP.DIR;1                                               6  24-SEP-1994 19:31
[]EXAMPLES.DIR;1                                             1  24-SEP-1994 19:31
[]SYSUPD.DIR;1                                               4  24-SEP-1994 19:31
[]SYSMSG.DIR;1                                               3  24-SEP-1994 19:31
   .
   .
   .
[]SECURITY_AUDIT.AUDIT                                       2   3-FEB-1995 15:23
[]SECURITY_AUDIT.AUDIT                                      11   3-FEB-1995 15:23
[]BACKUP.EXE;33                                            273   4-FEB-1995 09:37
[]STABACKUP.EXE;9                                          486   4-FEB-1995 09:38
```

If the display lists the files in the VMS$COMMON directory as lost files (files with an empty directory specification as shown in this example), your system disk is affected by this problem. Follow the procedure described above to correct it.

3.5 Batch and Print Queue Management

The following notes contain information about managing batch and print queues.

3.5.1 Changes and Enhancements

The following section describes a change pertaining to batch and print queue management.

3.5.1.1 Process Rights Identifier Limit Raised to 1024

V7.0 In OpenVMS Version 7.0, the maximum number of rights identifiers permitted for a process using the queue system has been raised to match the system limit of 1024.

3.5.2 Problems and Restrictions

The following sections describe problems and restrictions pertaining to batch and print queue management. For problems important to programmers, refer to Section 4.1.1.

3.5.2.1 PRINT/DELETE Command

VAX V5.5
Alpha V1.5

Before OpenVMS VAX Version 5.5 and OpenVMS Alpha Version 1.5, the queue manager allowed users to specify the PRINT/DELETE command for a file residing on a disk that was not mounted clusterwide, as long as the queue specified in the command was assigned to a node with access to the file being printed.

Starting with OpenVMS VAX Version 5.5 and OpenVMS Alpha Version 1.5, the new clusterwide queue manager process must have access to the file specified with the PRINT/DELETE command. Otherwise, the file is printed but not deleted.

This problem will be addressed in a future release of the operating system. Until then, you can ensure that the PRINT/DELETE command deletes the specified files by mounting the disks on which the files reside clusterwide. To mount a disk clusterwide, use the /CLUSTER qualifier with the MOUNT command.

However, if your operating environment does not allow you to mount a disk clusterwide, you can resolve this problem by running the queue manager process on a node that has access to the disk. You can specify the node on which the queue manager process runs by specifying the /ON=*node-list* qualifier with the START/QUEUE/MANAGER command. For more information on this qualifier, see the *OpenVMS DCL Dictionary*.

3.5.2.2 SUBMIT/DELETE Command

VAX V5.5
Alpha V1.5

The information in Section 3.5.2.1 also applies to the SUBMIT/DELETE command.

3.6 Compatibility Between VAX and Alpha Systems

The notes in this section contain information of interest to system managers who deal with both OpenVMS VAX and OpenVMS Alpha systems.

3.6.1 Documentation Changes and Corrections

The following note describes changes in documentation used by system managers who deal with both VAX and Alpha systems.

3.6.1.1 New Manuals Replace Old Ones

V7.0 The following manuals have been retired:

- *OpenVMS Compatibility Between VAX and Alpha*
- *A Comparison of System Management on OpenVMS AXP and OpenVMS VAX*

Much of the information that was contained in these manuals is included in two new migration manuals that ship with OpenVMS Version 7.0:

- *Migrating an Environment from OpenVMS VAX to OpenVMS Alpha*
- *Migrating an Application from OpenVMS VAX to OpenVMS Alpha*

The new manuals also include information from two other retired manuals:

- *Migrating to an OpenVMS AXP System: Planning for Migration*
- *Migrating to an OpenVMS AXP System: Recompiling and Relinking Applications*

3.7 DCL Commands

This section contains information about DCL commands commonly used by system managers.

3.7.1 Changes and Enhancements

This section describes changes and enhancements to DCL commands commonly used by system managers.

3.7.1.1 SHOW NETWORK Command: New Behavior

V7.0 The SHOW NETWORK command has been updated to display information about all registered network products. For example:

```
$ SHOW NETWORK
Product:  DECNET        Node:  MYNODE      Address(es):  63.639
$
```

You can still access the old DECnet SHOW NETWORK display by using the /OLD qualifier:

```
$ SHOW NETWORK/OLD

OpenVMS Network status for local node 63.639 MYNODE on 17-MAY-1995 19:51:33.40

This is a nonrouting node and does not have any network information.
```

See SYSTARTUP_VMS.TEMPLATE for examples for setting up your system to use these new network services.

If you are a provider of a network service and would like to take advantage of this new feature, you must register your product by placing a procedure SYS$NET_SERVICES_<product>.COM in SYS$STARTUP: that issues the correct SET NETWORK command. See the example file SYS$EXAMPLES:SYS$NET_SERVICES_EXAMPLE.COM.

3.7.1.2 ANALYZE/ERROR_LOG/SUMMARY Command

V6.2 The /SUMMARY qualifier is no longer supported for the ANALYZE/ERROR_LOG command.

3.7.2 Documentation Changes and Corrections

The following section provides updated descriptions of several DUMP command qualifiers used when dumping ISO 9660 volumes. For related information about mounting ISO 9660 volumes using SVDs, see Section 3.25.3.1.

3.7.2.1 *OpenVMS DCL Dictionary*

V7.0 The following corrections have been made to online Help and the Version 7.0 documentation that ships on the CD–ROM. Please update your hardcopy manual accordingly.

In the DUMP command documentation, replace the current descriptions of the /DESCRIPTOR, /MEDIA_FORMAT, and /PATH_TABLE qualifiers with the following descriptions:

Qualifiers

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

Dumps the specified ISO 9660 volume descriptors in a formatted manner. If /NOFORMATTED is specified, block mode format is used.

The descriptor options that you can specify are as follows:

BOOT:*n*	Searches for the *n*th occurrence of a Boot Record.
PVD:*n*	Searches for the *n*th occurrence of a Primary Volume Descriptor.
SVD:*n*	Searches for the *n*th occurrence of a Supplementary Volume Descriptor.
VPD:*n*	Searches for the *n*th occurrence of a Volume Partition Descriptor.
VDST:*n*	Searches for the *n*th occurrence of a Volume Descriptor Set Terminator.

If you specify only one option, you can omit the parentheses.

ISO 9660 descriptors are specified by their ordinal position from the start of the volume, defaulting to 1 if they are not specified. The ISO 9660 volume is sequentially searched from the beginning of the volume descriptor set sequence to the end to find the specified descriptor and output it in a formatted manner.

/MEDIA_FORMAT=keyword

Specifies the format in which a data structure is to be dumped. If you specify this qualifier, you must use one of the following keywords:

CDROM	Specifies ISO 9660 media format. This format is the default if you do not specify the /MEDIA_FORMAT qualifier.
CDROM_HS	Specifies High Sierra media format.

/PATH_TABLE

Dumps data blocks in ISO 9660 Path Table format.

3.8 Debugger

This section describes system management considerations for the OpenVMS Debugger in this release.

3.8.1 Problems and Restrictions

The following notes describe problems and restrictions in using the OpenVMS Debugger.

3.8.1.1 DECwindows Motif Requirement

V7.0 The DECwindows Motif interface to the OpenVMS Debugger Version 7.0 requires Version 1.2 or later of DECwindows Motif.

3.8.1.2 Displaying a Debug Session from a Personal Computer (PC)

V7.0 Although displaying a debug session from a PC is not officially supported or tested at this time, many users have reported successful results when using the following configurations:

- OpenVMS Debugger Version 6.2 or greater (earlier versions had some geometry problems)
- X-Windows Emulators:
  - Under Microsoft Windows: eXcursion Version 1.2A-1 (Win16)
  - Under MS–DOS: PC DECwindows Motif Version 5.1.005 (also known as DWDOS)
- Transports:
  - DEC TCP/IP Services using UCX Version 3.1 or 3.2
  - PATHWORKS Version 4.1 or 4.2
  - Windows Sockets TCP/IP Version 1.1

Several combinations of VAXstations, AlphaStations, 486, and Pentium processors have been tried, with no serious restrictions or performance problems.

Following are some general tips:

- Use a high-resolution monitor. A typical 640-X-480 pixel VGA display is too small. You can fine-tune the geometry and placement of your OpenVMS Debugger windows by editing your VMSDEBUG.DAT file (using the new customization features described in the debugger section of the *OpenVMS Version 7.0 New Features Manual*). It is suggested that you make the windows as small as you are comfortable with, and overlap their placement as needed to minimize display space requirements.
- Users of DECterm emulators should restrict their use to the screen-mode command interface, and not try to run the DECwindows Motif interface. Screen-mode debugging should be successful with any reasonably X-compliant terminal emulator, such as a DECterm (that is, created with the $CREATE /TERM/DETACH command).

3.8.1.3 Resource File for Customization and Internationalization

V7.0 The OpenVMS Debugger Version 7.0 provides improved user customization and internationalization features for the DECwindows Motif interface. There is a new version of the resource file, VMSDEBUG.DAT. See the sections on customization and internationalization in the debugger section of the *OpenVMS Version 7.0 New Features Manual* for information.

3.8.1.4 System Resources

V7.0

The kernel and main debugger communicate through global sections. The main debugger communicates with up to eight kernel debuggers through a 65-page (VAX) or 65-pagelet (Alpha) global section (65*512 bytes on either platform). Therefore, you might need to increase the system parameters GBLPAGES and GBLSECTIONS. For example, if 10 users use the debugger simultaneously, the debugger requires 10 global sections using a total of 650 global pages or pagelets.

3.8.1.5 User Quotas

V7.0

Each user needs a PRCLM quota sufficient to create an additional subprocess for the debugger, beyond the number of processes needed by the program.

BYTLM, ENQLM, FILLM, and PGFLQUOTA are pooled quotas. You may need to increase these quotas to account for the debugger subprocess as follows:

- You should increase each user’s ENQLM quota by at least the number of processes being debugged.
- You might need to increase each user’s PGFLQUOTA. If a user has an insufficient PGFLQUOTA, the debugger may fail to activate or may cause "virtual memory exceeded" errors during execution.
- You might need to increase each user’s BYTLM and FILLM quotas. The debugger requires sufficient BYTLM and FILLM quotas to open each image file being debugged, the corresponding source files, and the debugger input, output, and log files. To increase these quotas, system managers can run SYS$SYSTEM:AUTHORIZE.EXE. This adjusts parameters in SYSUAF.DAT.

3.9 DECamds

This section contains release notes pertaining to DECamds. New features of DECamds Version 7.0 are described in the *OpenVMS Version 7.0 New Features Manual*.

3.9.1 Changes and Enhancements

This section describes changes and enhancements to DECamds on OpenVMS systems.

3.9.1.1 Old VAX Versions Not Supported (VAX Only)

V7.0

VAX

Support for OpenVMS VAX Version 5.5 and OpenVMS VAX Version 6.0 has been removed. The minimum operating system that DECamds supports is OpenVMS Version 6.1. ♦

3.9.1.2 New Error Message

V7.0

The following new message has been added:

VERUNSUP, unsupported version on node 'node', must be at least V6.1

Explanation: The DECamds Data Analyzer has determined that the node to be monitored is running an older or unsupported version of the operating system.

User Action: Upgrade the remote node to the correct version.

3.9.2 Problems and Restrictions

The following sections describe known problems and restrictions in DECamds.

3.9.2.1 DECwindows Motif Support

V7.0 If you run DECwindows Motif Version 1.2, the color in the displays is not correct. You must install DECwindows Motif Version 1.2-3.

3.9.2.2 Kernel Threads Not Supported (Alpha Only)

V7.0

Alpha Support for kernel threads has not been implemented. If you use threaded processes, DECamds displays only the top thread. ♦

3.9.3 Corrections

The following sections describe DECamds problems that have been fixed in Version 7.0.

3.9.3.1 Unique Identification of Disks in a VMScluster

V7.0 The DECamds Disk Analyzer had problems in uniquely identifying a disk or volume across a cluster membership. (These problems were seen when large numbers of disk devices were attached to the cluster.) These problems have been fixed.

3.9.3.2 Image Name in Single Process Summary

V7.0 Because of a change in the way OpenVMS Alpha systems reference process page table entries, DECamds has been changed to modify the way image names are collected for this display.

Formerly, if the image name was not resident in memory, DECamds did not attempt to fetch the image name. Now DECamds attempts to obtain the image name of a process even if DECamds must request the process to perform a page fault to obtain the data. A single page fault should not be harmful for long-running processes and now the data can be available for processes that run many images.

To use the new algorithm, the new Data Provider must be loaded onto the system where the data is to be collected.

VAX For VAX systems, this involves simply installing the new Data Provider and entering the following command:

```
$ @SYS$STARTUP:AMDS$STARTUP.COM RELOAD ♦
```

Alpha For Alpha systems with DECamds already installed, you must install the new version of the Data Provider and then reboot. OpenVMS Alpha systems do not support the RELOAD option. ♦

3.9.3.3 Saving Resource Hash Table Dense/Sparse Thresholds

V7.0 The thresholds for the Resource Hash Table Dense/Sparse Thresholds are now saved correctly.

3.10 DECdtm Services in a DECnet/OSI Network

This section contains notes pertaining to DECdtm services.

3.10.1 Problems and Restrictions

This section describes known problems and restrictions associated with using DECdtm services.

3.10.1.1 Assigning SCSNODE Values

V6.1 Read this section if all of the following apply to you:

- You have a DECnet/OSI network.
- You use DECdtm services.
- Your DECdtm transactions span different VMScluster systems or standalone computers.

SCSNODE is a system parameter that defines the name of a computer. DECdtm transactions can fail if the same SCSNODE value is used for more than one computer.

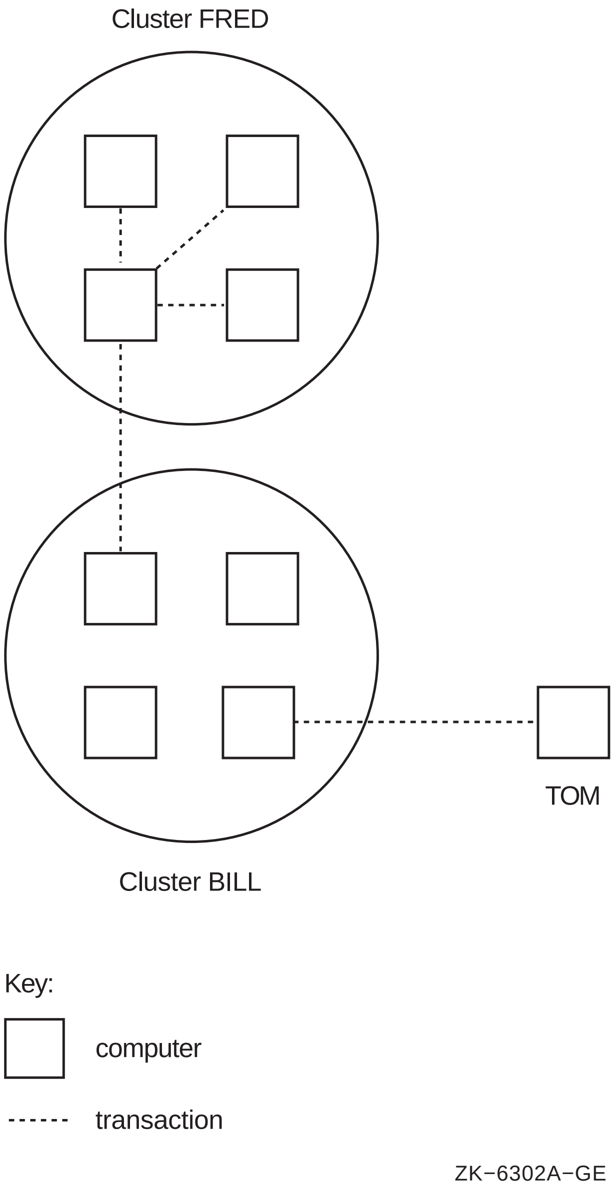
Make sure the same SCSNODE name is not duplicated on other computers by following these steps:

1. Make a note of the computers that belong to your **transaction group**. A transaction group is a group of computers involved in DECdtm transactions, where:
   - A computer belongs to only one transaction group.
   - Every computer in a VMScluster belongs to the same transaction group.
   - Computers A and B belong to the same transaction group if any transaction on computer A involves computer B.
   - Computers A and C belong to the same transaction group if any transaction on computer A involves computer B, and any transaction on computer B or any node in B's VMScluster involves computer C.

   Figure 3–1 shows an example of a transaction group.
2. For each computer in your transaction group, make sure that the SCSNODE value is different from:
   - The SCSNODE values of other computers in the transaction group
   - DECnet synonyms of other computers in the entire network
   - DECnet simple names of other computers on the same local root
3. If the computer is part of a VMScluster, also make sure that the SCSNODE value is different from:
   - DECnet simple names of other computers in the same VMScluster
   - DECnet simple names of computers on the same local root as other VMScluster members

For information on how to find out DECnet synonyms and DECnet simple names, see the *DECdns Management* manual. For information on how to find out or change the SCSNODE name, see the *OpenVMS System Manager's Manual*.

Figure 3–1 Transaction Group



Note the following in the example in Figure 3–1:

- Transactions on a computer in cluster FRED involve other computers in cluster FRED and a computer in cluster BILL.
- Transactions on a computer in cluster BILL involve standalone machine TOM.
- No other computers in the network are involved in transactions with computers in clusters FRED or BILL, or with standalone computer TOM.

Therefore, the computers in the transaction group are:

All computers in cluster FRED
All computers in cluster BILL
Computer TOM

3.11 DECevent Fault Management Support (Alpha Only)

V6.2

Alpha

Starting with OpenVMS Alpha Version 6.2, DECevent replaces the Error Reporting Formatter (ERF) as the bit-to-text translating tool for fault management on OpenVMS Alpha systems. While the ERF is still available on Version 6.2, it will be retired in a future release of the OpenVMS Alpha operating system.

3.11.1 Problems and Restrictions

This section describes problems and restrictions related to using the DECevent fault management tool.

3.11.1.1 Bit-to-Text Translation Support

V6.2

DECevent bit-to-text translation is supported on many products and devices. On other devices, as much translation as possible is performed and all remaining information in the event is dumped in hexadecimal.

Contact your Digital support representative if you have questions about the type of DECevent support currently available for your devices.

3.11.1.2 Logical File Names

V6.2

DECevent is unable to translate as input any logical defined as a search list of file names. For example:

```
$define event_log disk1:[events]event_log1.sys,disk1:[events]event_log2.sys
$dia/analyze event_log

DECevent T1.0 FT2
_DIAGNOSE-FAT: Analyze - No files found ' event_log '
_DIAGNOSE-FAT: An error occurred while executing a command ruleset
_DIAGNOSE-INF: No Error Messages to send in thread 1
```

3.11.2 Documentation Changes and Corrections

The following note describes a change in the DECevent documentation.

3.11.2.1 Documentation Location

V7.0

For Version 7.0 of OpenVMS, DECevent documentation is included on the Documentation CD–ROM in PostScript (.PS) and Bookreader (.DECW$BOOK) formats. Refer to the *Guide to OpenVMS Version 7.0 CD–ROMs* for more information about DECevent documentation and file names. ♦

3.12 DECnet for OpenVMS Alpha (Phase IV)

Alpha

The following sections describe new support and restrictions that apply to DECnet for OpenVMS Alpha (Phase IV). For information about using DECnet/OSI (Phase V), see the DECnet/OSI documentation.

3.12.1 Problems and Restrictions

V1.5

The following restrictions apply to DECnet for OpenVMS Alpha:

- Level 1 routing is available but is supported only on DECnet for OpenVMS Alpha nodes acting as routers for a cluster alias. Routing between multiple circuits or having more than one routing circuit enabled at once is not supported.
- Level 2 routing is not supported on DECnet for OpenVMS Alpha nodes.

- Some line types are unsupported.

  DECnet for OpenVMS Alpha nodes can connect to a DECnet network only via Ethernet lines or FDDI lines. DECnet communication over CI lines is not supported. There also is no support for DDCMP lines.

  Because DDCMP lines are unsupported, the DCL command SET TERMINAL /PROTOCOL=DDCMP /SWITCH=DECNET also is unsupported on Alpha systems. ♦

3.13 DECnet for OpenVMS VAX (Phase IV)

VAX

The following sections describe new support and restrictions that apply to DECnet for OpenVMS VAX (Phase IV). For information about using DECnet/OSI (Phase V), see the DECnet/OSI documentation.

3.13.1 Changes and Enhancements

The following section describes changes to DECnet for OpenVMS VAX (Phase IV).

3.13.1.1 DEFQA Now Can Be Configured with SYS$MANAGER:NETCONFIG.COM

V6.2

The DEC FDDIcontroller/Q–bus Controller (DEFQA) is now supported by DECnet for OpenVMS VAX (Phase IV).

The DEFQA can now be configured using the DECnet for OpenVMS VAX NETCONFIG.COM procedure. The NCP LINE and CIRCUIT name for the DEFQA is FQA-*n*, where *n* is the number representing the controller; for example, line FQA-0 maps to device FQA0 and line FQA-1 maps to device FQB0.

If you previously had the following logical defined in SYS$MANAGER:SYCONFIG.COM in order to use this device on OpenVMS VAX Version 6.1, you must remove the logical from SYCONFIG.COM and deassign it before configuring the device. This logical is no longer necessary to configure the DEFQA.

```
$ DEFINE/SYSTEM FXc0 FQc0:
```

3.13.2 Problems and Restrictions

The following section describes a restriction pertaining to DECnet for OpenVMS VAX (Phase IV).

3.13.2.1 DEFTA Cannot Be Configured with SYS$MANAGER:NETCONFIG.COM

V6.1

DECnet for OpenVMS VAX cannot configure the DEC FDDIcontroller/TURBOchannel Controller (DEFTA).

To use DECnet for OpenVMS with the DEFTA on OpenVMS VAX Version 6.1 and later, you must define a logical name before invoking the NETCONFIG.COM or STARTNET.COM command procedures. Use the following DCL command to define the required logical name:

```
$ DEFINE/SYSTEM FZc0 FCc0:
```

In this command, *c* represents the controller (for example, FCA0).
To make the definition permanent, define this logical name in SYS$MANAGER:SYCONFIG.COM.

The NCP LINE and CIRCUIT name for the DEFTA controller is FZA-*n*, where *n* is the number representing the controller; for example, line FZA-0 maps to device FZA0 and line FZA-1 maps to device FZB0. ♦

3.14 DECnet/OSI for OpenVMS Alpha (Phase V)

Alpha

The following sections contain notes that apply to DECnet/OSI for OpenVMS Alpha (Phase V). For complete information about using DECnet/OSI, see the DECnet/OSI documentation.

3.14.1 Problems and Restrictions

The following section describes a restriction in DECnet/OSI for OpenVMS Alpha.

3.14.1.1 MOP Incompatible With Kernel Threads

V7.0

In order to use DECnet/OSI maintenance operation protocol (MOP) on an Alpha Version 7.0 system, you must set the SYSGEN parameter MULTITHREAD to 0. This 0 setting also disables kernel threads on the system.

This restriction will be removed in a future release. ♦

3.15 DECnet/OSI for OpenVMS VAX (Phase V)

VAX

The following sections contain notes that apply to DECnet/OSI for OpenVMS VAX (Phase V). For complete information about using DECnet/OSI, see the DECnet/OSI documentation.

3.15.1 Changes and Enhancements

The following section describes changes to DECnet/OSI.

3.15.1.1 DEFQA Now Can Be Configured with SYS$MANAGER:NET$CONFIGURE.COM

V6.2

The DEC FDDIcontroller/Q–bus Controller (DEFQA) is now supported by DECnet/OSI for OpenVMS VAX (Phase V).

The DEFQA can be configured by DECnet/OSI for OpenVMS VAX using the NET$CONFIGURE.COM procedure. For information on how to configure devices using DECnet/OSI, refer to the *DECnet/OSI for OpenVMS Installation and Configuration* manual.

If you previously had the following logical defined in SYS$MANAGER:SYCONFIG.COM in order to use this device on OpenVMS VAX Version 6.1, you must remove the logical from SYCONFIG.COM and deassign it before configuring the device. This logical is no longer necessary to configure the DEFQA.

```
$ DEFINE/SYSTEM FXc0 FQc0:
```

3.15.1.2 DEFTA Now Can Be Configured with SYS$MANAGER:NET$CONFIGURE.COM

V6.2

The DEC FDDIcontroller/TURBOchannel Controller (DEFTA) is now supported by DECnet/OSI for OpenVMS VAX (Phase V).

The DEFTA can be configured by DECnet/OSI for OpenVMS VAX using the NET$CONFIGURE.COM procedure. For information on how to configure devices using DECnet/OSI, refer to the *DECnet/OSI for OpenVMS Installation and Configuration* manual.

If you previously had the following logical defined in SYS$MANAGER:SYCONFIG.COM in order to use this device on OpenVMS Version 6.1, you must remove the logical from SYCONFIG.COM and deassign it before configuring the device. This logical is no longer necessary to configure the DEFTA using DECnet/OSI.

```
$ DEFINE/SYSTEM FZc0 FCc0: ♦
```

3.16 DECwindows Motif for OpenVMS

This section contains release notes pertaining to the DECwindows Motif for OpenVMS layered product.

3.16.1 Changes and Enhancements

This section includes notes about support for the the DECwindows Motif for OpenVMS layered product.

3.16.1.1 DECwindows Motif Version 1.1 for OpenVMS No Longer Supported

V7.0

Starting with OpenVMS Version 7.0, the OpenVMS operating system no longer supports DECwindows Motif Version 1.1 for OpenVMS. To use DECwindows with OpenVMS Version 7.0 and later, you must install DECwindows Motif Version 1.2 or later. However, DECwindows Motif Version 1.2 and later provide run-time support for programs built on earlier versions of DECwindows and DECwindows Motif. For more information, see the *DECwindows Motif for OpenVMS Release Notes* for Version 1.2 or later. See Section 3.16.2.1 for a Version 1.2 restriction.

3.16.1.2 NCSA Mosaic Browser: TCP/IP Support Options

V7.0

DECwindows Motif Version 1.2–3 includes the NCSA Mosaic Version 2.4 web browser. The *DECwindows Motif Version 1.2–3 for OpenVMS Release Notes* state that the NCSA Mosaic browser requires that the TCP/IP Services for OpenVMS product (UCX) be installed on your system. Note that TCP/IP support can alternatively be provided by PathWay for OpenVMS (Wollongong), TCPware (Process Software), or MultiNet Version 3.4B (TGV).

3.16.2 Problems and Restrictions

This section describes problems and restrictions associated with the DECwindows Motif for OpenVMS layered product.

3.16.2.1 DECwindows Motif Version 1.2 Compatibility Problem With Version 7.0

V7.0

The DECwindows Motif Version 1.2 will not run on OpenVMS Version 7.0 systems. To correct this problem, users must install one of the following remedial kits:

- For Alpha systems, AXPMOTF05_012
- For VAX systems, VAXMOTF05_012

Alternatively, users can install DECwindows Motif Version 1.2–3.

3.16.2.2 DEC Notes Fails With LinkWorks Error Message (Alpha Only)

V7.0

Alpha

If you run DECwindows Motif Version 1.2–3 for OpenVMS on an OpenVMS Alpha Version 7.0 system and use the DECwindows interface to DEC Notes, Notes fails to open remote notes conferences and displays the following error message:

```
 LinkWorks Reported Error: Unknown error
```

To correct this problem, install patch kit AXPMOTF06_U3012, which is on the OpenVMS Version 7.0 CD–ROM in the same directory as the DECwindows Motif V1.2-3 for OpenVMS Alpha kit. ♦

3.16.2.3 Console Broadcasts Disabled

V6.2

In DECwindows Motif Version 1.2–3, console broadcasts are disabled by default by the DECwindows startup procedure on nonworkstation systems as well as on workstation systems. To allow broadcasts to OPA0:, edit the file SYS$MANAGER:DECW$PRIVATE_APPS_SETUP.COM (creating it if it does not exist) and add the following global symbol definition:

```
$ DECW$CONSOLE_SELECTION == "ENABLE"
```

Then, restart DECwindows by entering the following command:

```
$ @SYS$MANAGER:DECW$STARTUP RESTART
```

On workstation systems, Digital recommends that you set DECW$CONSOLE_SELECTION to WINDOW instead of ENABLE. This directs console output to a Console Window application, which is new in DECwindows Motif V1.2–3, instead of to the operator window on the graphics screen.

If you prefer that console broadcasts not be disabled by default on nonworkstation systems, install the following remedial kits, which are available through TIMA unless otherwise noted:

- For VAX systems, VAXMOTF05_U3012
- For Alpha systems, AXPMOTF05_U3012 or AXPMOTF06_U3012 (available on the Version 7.0 CD–ROM only)

Note that console broadcasts will still be disabled by default on workstations.

3.16.2.4 System Files Purged During Startup

V6.2

In Version 1.2–3, the following DECwindows files are purged each time DECwindows Motif starts:

SYS$LIBRARY:DECW$*.EXE
SYS$SYSTEM:DECW$SETSHODIS.EXE

This problem is corrected by installing the following remedial kits, which are available through TIMA unless otherwise noted:

- For VAX systems, VAXMOTF05_U3012
- For Alpha systems, AXPMOTF05_U3012 or AXPMOTF06_U3012 (available on the Version 7.0 CD–ROM only)

3.16.2.5 Virtual Data Error Message (Alpha Only)

V6.2

Alpha

In Version 1.2–3, the following error message is displayed on nonworkstation OpenVMS Alpha systems during system startup:

```
%SDA-E-NOTINPHYS, 00000024 : virtual data not in physical memory
```

You can ignore this message. ♦

This problem is corrected by installing the following remedial kits, which are available through TIMA unless otherwise noted:

- For VAX systems, VAXMOTF05_U3012
- For Alpha systems, AXPMOTF05_U3012 or AXPMOTF06_U3012 (available on the Version 7.0 CD–ROM only)

3.16.2.6 World Wide Support Requires Remedial Kit

V6.2

Before you upgrade to OpenVMS Version 6.2 and later on systems that run language variant versions of DECwindows Motif Version 1.2, you should first install a remedial kit for DECwindows Motif Version 1.2 World Wide Support. This patch is necessary because the internationalization support provided in OpenVMS Version 6.2 and later is incompatible with the support in DECwindows Motif Version 1.2.

The name of the remedial kit is DWMV_I18N01_012 on VAX and DWMA_I18N01_012 on Alpha. If you install OpenVMS Version 6.2 and later from the CD–ROM, the remedial kit will be on the CD–ROM; otherwise you must obtain the kit from your Digital support representative.

Note

Language variants are now available in DECwindows Motif Version 1.2–3. If you are running DECwindows Motif Version 1.2–3, you do not need to install a remedial kit.

3.16.2.7 DECwindows Server Height or Width Exceeding 32767 (VAX Only)

V6.1

VAX

When an X application sends the display server a width or height greater than 32767, the application may terminate with a BadValue error similar to the following:

```
X error event received from server: BadValue (integer parameter out of
range for operation)
  Major opcode of failed request: 61 (X_ClearArea)
  Value in failed request: 0xffff****
  Serial number of failed request: ###
  Current serial number in output stream: ###
```

The following calls can cause this problem:

CopyArea()
CreateWindow ()
PutImage()
GetImage()
CopyPlane()
ClearArea()

This is due to the width and height being defined as a signed word by the display server when it should be defined as an unsigned word (CARD16) that allows for values up to 65536.

To modify the default operation perform the following steps:

1. Set the logical name DECW$CARD16_VALIDATE to TRUE as follows:

```
$DEFINE/TABLE=DECW$SERVER0_TABLE   DECW$CARD16_VALIDATE  TRUE
```

2. Exit the session and log back in.

   Exiting the session causes the display server to reset using the new value of the logical name DECW$CARD16_VALIDATE. The server will now accept values that are greater then 32767 without generating an error.

To make this a permanent change, add the command from step 1 to the file SYS$MANAGER:DECW$PRIVATE_SERVER_SETUP.COM.

3.16.2.8 SET DISPLAY Used to Create WSA Pseudo Workstation Devices

V6.1

When creating WSA pseudo workstation devices using the SET DISPLAY command, be careful not to create WSA devices that are never destroyed. For example, this DCL command procedure is wrong:

```
$LOOP:
$  SET DISPLAY/CREATE/NODE=remote
$  RUN SYS$SYSTEM:DECW$CLOCK
$  IF $STATUS THEN GOTO DONE
$  WAIT 0:0:5
$  GOTO LOOP
$DONE:
```

If the clock cannot be started for some reason, one WSA device will be created for each failed attempt. These WSA devices will use up non-paged dynamic memory, and eventually the process will exceed its BYTLM quota and enter a resource wait state (if resource waiting is enabled, as it is by default).

A better version of this command procedure is the following:

```
$  SET DISPLAY/CREATE/NODE=remote
$LOOP:
$  RUN SYS$SYSTEM:DECW$CLOCK
$  IF $STATUS THEN GOTO DONE
$  WAIT 0:0:5
$  GOTO LOOP
$DONE:
$  SET DISPLAY/DELETE ’F$TRNLNM("DECW$DISPLAY")’
```

The SET DISPLAY/DELETE command deletes the WSA device that was created at the beginning of the command procedure; the logical name DECW$DISPLAY contains the name of the WSA device that was created.

3.17 Digital Distributed Computing Environment (DCE) for OpenVMS

This section contains notes pertaining to the layered product, Digital Distributed Computing Environment (DCE) for OpenVMS VAX and OpenVMS Alpha.

3.17.1 Problems and Restrictions

The following note describes support restrictions.

3.17.1.1 Support Restrictions

V7.0

To use Digital DCE for OpenVMS with OpenVMS Version 7.0, you must upgrade Digital DCE for OpenVMS to Version 1.3B.

Alpha

On Alpha systems, Digital DCE for OpenVMS Version 1.3B works only with kernel threads disabled. To disable kernel threads, set the SYSGEN parameter MULTITHREAD to 0.

When installing Digital DCE for OpenVMS on OpenVMS Alpha Version 7.0, you might get the following error message:

```
%INSTALL-E-FAIL, failed to CREATE entry for DTSS$RUNDOWN.EXE
-INSTALL-E-SYSVERDIF, system version mismatch - please relink
```

You can ignore this error. The correct version of DTSS$RUNDOWN.EXE is installed by Digital DCE for OpenVMS. ♦

3.18 Help Message Utility (MSGHLP)

This section contains release notes pertaining to the Help Message utility (MSGHLP).

3.18.1 Problems and Restrictions

This section describes problems and restrictions related to using the Help Message utility.

3.18.1.1 User-Supplied Comments Are Not Preserved

VAX V6.0
Alpha V1.0

Currently, user-supplied comments or additions to Digital-supplied .MSGHLP$DATA files will not be preserved through the next upgrade. However, your own .MSGHLP$DATA files are not affected by upgrades.

Note that you can reuse .MSGHLP files to insert your own messages into future Digital-supplied database files. Depending on the content of future databases, you might also be able to reuse some .MSGHLP files to add comments to existing messages.

3.19 Image Registration

This section contains notes pertaining to image registration.

3.19.1 Problems and Restrictions

The following note describes a restriction when image registration is run for shareable images on OpenVMS Alpha.

3.19.1.1 Shareable Images Must Be Installed With /OPEN (Alpha Only)

V7.0

Alpha

In OpenVMS Alpha Version 7.0, image registration works correctly for shareable images only if the image is installed using the /OPEN qualifier; for example:

```
$ INSTALL ADD SYS$LIBRARY:SNASHR /OPEN
```

This restriction will be lifted in a future release of OpenVMS Alpha. ♦

3.20 INITIALIZE

This section describes notes pertaining to INITIALIZE.

3.20.1 Problems and Restrictions

The following section describes an INITIALIZE problem.

3.20.1.1 Recovery of Badly Initialized Root Directory (Alpha Only)

V6.1

Alpha

In the Software Developer Toolkit (SDK) version of OpenVMS Alpha Version 6.1, INITIALIZE did not initialize the root directory of a volume, 000000.dir correctly. This bad initialization could cause the directory entry for this file to be deleted, and also prevent you from being able to set ACLs on 000000.dir.

To tell whether you have a disk that has suffered from this, use the ANALYZE /DISK_STRUCTURE command on the disk. A new warning message has been added to flag this condition:

```
%ANALDISK-W-BADINITD_MFD, Root directory 000000.DIR;1 file header
incorrectly initialized, RVN 1
```

Use the ANALYZE/DISK_STRUCTURE/REPAIR command to repair this bad initialization.

If the directory entry for 000000.dir has been deleted, then use ANALYZE/DISK_STRUCTURE/REPAIR to repair this as well. You will receive the following warning from ANALYZE/DISK_STRUCTURE if this directory entry has been deleted:

```
%ANALDISK-W-LOSTHEADER, file (4,4,1) 000000.DIR;1 not found in a directory ♦
```

3.21 Install Utility

This section contains release notes pertaining to the Install utility (INSTALL).

3.21.1 Changes and Enhancements

This section describes a change to INSTALL.

3.21.1.1 Installing Images

V6.2 The REPLACE option for the OpenVMS Install utility (INSTALL) has been changed to modify the known file database in an atomic fashion.

In the past, REPLACE was equivalent to DELETE followed by ADD. Consequently, there was a short time during which neither the new nor the old image was in the known file database. When activating protected or privileged images, this could result in failed image activations. Also, if the new image could not be installed, it was possible for neither the old nor the new image to be installed after the failure.

These problems are now corrected.

With the change, REPLACE operations for images installed with the /SHARED qualifier might require more global sections or global pages than in the past. Also, the names of global sections have been changed to avoid naming conflicts. The global sections can be displayed with one of the following commands:

```
$ INSTALL LIST /GLOBAL
```

or

```
$ INSTALL LIST image-name /GLOBAL
```

3.22 Mail Utility

This section contains notes of interest to system managers regarding the Mail utility. Notes of general interest are documented in Section 2.8.

3.22.1 Changes and Enhancements

The following note describes an important change to Mail.

3.22.1.1 MAIL.EXE and Privileges

V7.0 The OpenVMS installation procedure does not initially install MAIL.EXE with any privileges (because MAIL.EXE does not require privileges to perform its functions). Prior versions of the OpenVMS operating system did include mechanisms that allowed MAIL.EXE to check, ignore, grant, or override certain privileges that a system manager might assign when reinstalling MAIL.EXE. Because these regulatory mechanisms sometimes created unexpected or undesirable conditions, they have been removed in Version 7.0 of the OpenVMS operating system.

Caution

If you reinstall MAIL.EXE with certain privileges, you must carefully consider possible ramifications, including the potential for security breaches. For example, because MAIL.EXE confers its privileges on any user who invokes the Mail utility, that user will inherit those privileges if the user creates a subprocess from within Mail by specifying the SPAWN command.

3.22.2 Problems and Restrictions

The following note describes how to fix a problem in Mail.

3.22.2.1 Replying to an Address Containing Nested Quotation Marks

V7.0 In some cases, you cannot use the Mail command REPLY to reply to mail received from an address containing nested quotation marks.

To correct this condition, you can set bit 5 (bitmask value 16) in the logical MAIL$SYSTEM_FLAGS. The example in SYS$MANAGER:SYLOGICALS.TEMPLATE shows how to do this.

3.23 Monitoring Performance History (MPH)

V7.0 Digital invites you to participate in the Digital Product Performance (DPP) Program, which monitors and verifies in-field performance of Digital systems at customer and Digital sites. This program provides Digital service, manufacturing, and design engineering organizations with accurate information about the performance of Digital products. Digital's goal is to provide customers with improved reliability on all Digital systems.

To ensure the high quality of its products, Digital has developed a system monitoring tool called Monitoring Performance History (MPH). MPH resides on participants' systems with negligible impact on system performance and no impact on system security.

MPH collects error log entries, crash dump summaries, and configuration information from monitored systems. Every week the information is sent to the DPP group using the selected transport mechanism, which can include DSNLink and Internet mail, among others.

DPP analyzes the collected information and generates reports that are distributed to engineering, manufacturing, and services groups. These groups evaluate this information in an effort to improve system reliability and availability. All collected information is classified as Digital Confidential and is held for internal use only.

The MPH process, with the exception of installation, is fully automated. MPH runs as a background task utilizing negligible system resources. The disk space required for the collected data and the application is approximately 300 blocks per system.

MPH is a voluntary program that requires no special maintenance agreement with Digital.

MPH Kit Location on the Media

The MPH kit and installation guide are in the following OpenVMS Version 7.0 media locations:

- Volume 3 of the magnetic tape media
- Volume 2 of the TK50 media
- Directory [MPH] of the CD–ROM media

Installing MPH

You install MPH by using VMSINSTAL. The installation manual is in the MPH kit and can be extracted in either text form or POSTSCRIPT form.

- To extract the installation manual in text form, enter the following command:

  ```
  $ BACKUP/SELECT=MPH_OVMS_INSTALL_GUIDE.TXT MPH_VMS013.A /SAVE []
  ```

- To extract the information in POSTSCRIPT form, enter the following command:

  ```
  $ BACKUP/SELECT=MPH_OVMS_INSTALL_GUIDE.PS MPH_VMS013.A /SAVE []
  ```

Stopping MPH

You can stop MPH on your systems at any time by entering this command:

```
$ @SYS$MANAGER:MPH$SHUTDOWN.COM
```

Deinstalling MPH

You can deinstall MPH at any time by entering this command:

```
$ @SYS$MANAGER:MPH$DEINSTAL.COM
```

New Features in Version 1.3 of the OpenVMS MPH Kit

The following new features are included in Version 1.3 of the OpenVMS MPH kit:

- MPH now offers you the opportunity to upgrade your existing installation.
- Error log data can be collected on a weekly or daily basis.
- Customer questions asked during the installation process have been updated to retrieve more accurate information.
- During installation, if Internet is chosen as the MPH file transport method, an MPH_INTERNET.DIS file is created which is used to mail the files back to DPP.
- MPH now performs space management to ensure it has no impact on your system, regardless of error log size.
- The MPH directory is no longer required to be a top-level directory.
- MPH invokes the CRash Log tool (CRL) on OpenVMS VAX systems if the CRash Log Utility Extractor (CLUE) is not running on the system.
- Several new safeguards and warning messages have been introduced; for example:
  - The system manager is notified if there are problems with DSNLink.
  - The system manager is notified if the initial error log is too large or if daily changes to the error log are large.
- After installation on a cluster, monitored nodes can be added or deleted by executing MPH$AREA:MPH$EDIT_NODES.COM.

For more information...

For additional information about MPH or the DPP program, send mail to:

```
mph_admin@dppsys.enet.dec.com
```

Include the keyword INFO anywhere in the subject header of your message to expedite routing of your request.

Digital looks forward to your participation in this mutually beneficial program. Thank you for your cooperation.

3.24 Monitor Utility

The following notes pertain to the Monitor utility (MONITOR).

3.24.1 Changes and Enhancements

This section describes changes and enhancements to MONITOR.

3.24.1.1 TCP/IP Support

V7.0

The Monitor utility has been enhanced to be able to use either TCP/IP (if available) or DECnet as its transport. MONITOR will try to access TCP/IP first; if TCP/IP is not available, MONITOR will use DECnet.

To take advantage of this enhancement, you must uncomment the following line in SYS$STARTUP:SYSTARTUP_VMS.COM:

```
$ @SYS$STARTUP:VPM$STARTUP.COM
```

See SYS$STARTUP:SYSTARTUP_VMS.TEMPLATE for examples.

3.24.1.2 Disk Limit Increased

V6.2

Beginning with OpenVMS Version 6.2, the limit on the number of disks that can be monitored was raised from 799 to 909 for record output and from 799 to 1817 for display and summary outputs. If you are monitoring a remote node running OpenVMS Version 6.2 or later from a system running a version earlier than OpenVMS Version 6.2, the old limit of 799 applies.

3.24.1.3 Display Header Changed From "VAX/VMS" To "OpenVMS" (VAX Only)

V6.2

VAX

The text in the first heading line for MONITOR display and summary output has been changed slightly. The heading line "VAX/VMS Monitor Utility" used in previous VAX versions now reads "OpenVMS Monitor Utility" to match OpenVMS Alpha. This change is a reflection of the functional equivalence of MONITOR on both platforms.

Any software that parses this line of the display or summary output of MONITOR should be changed accordingly. This change does not affect MONITOR's binary recording file output. ♦

3.25 Mount Utility

The following notes pertain to the Mount utility (MOUNT).

3.25.1 Problems and Restrictions

The following notes describe restrictions in using MOUNT.

3.25.1.1 MOUNT/FOREIGN Limitation

V7.0

Devices mounted using the MOUNT/FOREIGN command cannot undergo mount verification; therefore they are unable to fail over to an alternate I/O path in the event of a path failure.

3.25.1.2 ISO 9660 Support in Mixed-Version VMScluster Systems

V6.1

On OpenVMS systems, do not specify the /CLUSTER qualifier when mounting ISO 9660 formatted CD–ROMs in a VMScluster system with nodes that are running versions of OpenVMS VAX prior to Version 6.0. If you attempt to mount an ISO 9660 CD–ROM on an OpenVMS node without ISO 9660 support, which is likely in a mixed-version environment, the operation will fail. However, the failure will take an excessive amount of time to complete, due to the slow access time for the CD–ROM media.

3.25.2 Corrections

This section describes software corrections to MOUNT.

3.25.2.1 Noncompliant ISO 9660 CD–ROMs and MOUNT (VAX Only)

V6.2

VAX

Previously, on OpenVMS VAX systems, support was not provided for mounting noncompliant ISO 9660 CD–ROMs.

If users had noncompliant media and were running applications in a mixed-architecture VMScluster system (OpenVMS Alpha Version 6.1 with either OpenVMS VAX Version 6.0 or OpenVMS VAX Version 6.1), they could experience problems. This was an issue for PATHWORKS users who were using noncompliant ISO 9660 CD–ROMs in a mixed-architecture VMScluster system.

This problem has been corrected. ♦

3.25.3 Documentation Changes and Corrections

The following section provides information about mounting ISO 9660 volumes using SVDs. For related information about dumping ISO 9660 volumes, see Section 3.7.2.1.

3.25.3.1 *OpenVMS System Manager's Manual: Essentials*

V7.0

Insert the following section between Section 8.7.3 (Handling Partially Mounted ISO 9660 Volume Sets) and Section 8.7.4 (Handling ISO 9660 Restrictions):

Mounting ISO 9660 Volumes Using SVDs

In addition to mounting ISO 9660 volumes using the default Primary Volume Descriptor (PVD), you can also mount ISO 9660 volumes using a Supplementary Volume Descriptor (SVD).

This capability allows access to an ISO 9660 volume with directories and file names containing characters from character sets other than the ISO 9660 limited set, which includes only A through Z, "_" (underscore), "." (period), and ";" (semicolon).

The author of the ISO 9660 volume set must have recorded the volume with the required PVD, and optionally with one or more SVDs. Each SVD must contain a unique volume label and escape sequence.

Use the following command syntax to mount an ISO 9660 device using an SVD:

```
MOUNT device-name volume-label /UCS_SEQUENCE=escape-sequence
```

where:

device-name	Specifies the physical device name or logical name of the device on which the ISO 9660 volume is to be mounted.
volume-label	Specifies the SVD volume label obtained from the author's label on the CD–ROM.
escape-sequence	Specifies the escape sequence obtained from the author's label on the CD–ROM.

If an ISO 9660 volume contains SVDs with no escape sequence specified, the default character set is assumed to be ISO 646 (ASCII). This default character set allows the file specification character set supported by OpenVMS to be used, which includes these additional characters: "$" (dollar sign) and "-" (dash).

Use the following command syntax to mount a volume using the SVD volume label when no escape sequence is specified:

```
MOUNT device-name volume-label /UCS_SEQUENCE=""
```

Note

If an ISO 9660 volume contains SVDs with escape sequences other than ISO 646 (see ISO 2022), the character set used might not interoperate with the OpenVMS file specification syntax.

3.26 Nonpaged Pool

This section contains release notes pertaining to nonpaged pool.

3.26.1 Problems and Restrictions

This section describes software problems and restrictions related to nonpaged pool.

3.26.1.1 Prepopulation of Nonpaged Pool (VAX Only)

V6.2

VAX

Feedback from some sites suggests that prepopulation of the nonpaged pool lookaside lists can cause problems in some environments. Starting with Version 6.2 of OpenVMS VAX, the SYSGEN parameter VMS7 is used as a switch to turn off prepopulation.

If you experience any problems that you believe are related to nonpaged pool prepopulation, set VMS7 to 1. ♦

3.27 Operator Communication Manager

The release notes in this section pertain to the operator communication manager (OPCOM).

3.27.1 Problems and Restrictions

The release notes in this section describe restrictions you need to be aware of when using OPCOM.

3.27.1.1 OPERATOR.LOG File

V6.2 The size of and access to the OPERATOR.LOG file (or to the file pointed to by the logical OPC$LOGFILE_NAME) is limited by the size and access of the disk device on which it resides. If the disk device does not have enough room to write to the log file or if access to the device is restricted in any other way, records might be missing from the log file.

3.28 POLYCENTER Software Installation Utility

The notes in this section pertain to the POLYCENTER Software Installation utility. Also see Section 4.22 for notes about this utility that are of interest to programmers.

3.28.1 Problems and Restrictions

Notes in this section pertain to problems and restrictions with using the POLYCENTER Software Installation utility to install, remove, and reconfigure software products. Problems and restrictions of interest to programmers are described in Section 4.22.1.

3.28.1.1 PRODUCT Command

There are several problems with the DCL command PRODUCT:

- V6.2 Commands such as PRODUCT FIND and PRODUCT SHOW that can display more than a screen of text do not provide qualifiers such as /PAGE to control scrolling or /OUTPUT to redirect output to a file. As a result, information can scroll off the screen.
- V6.1 The /DIRECTORY and /DEVICE qualifiers do not work correctly on the PRODUCT SHOW OBJECT command. The utility either ignores them or provides output that does not match the qualifiers.
- The PRODUCT command can cause indexed and relative files to be truncated during a package or install operation. To determine if a particular file is susceptible to being truncated, use the DIRECTORY/SIZE=ALL command to examine the file. If the two sizes are the same, your file will not exhibit this problem. If the two sizes are not the same, perform the following SET FILE command prior to a package or install:

  ```
  $ SET FILE/END_OF_FILE file-in-question.dat
  ```

  This command should be issued only for an indexed or relative file, since the end-of-file (EOF) attribute is meaningful for sequential files.

Digital expects to correct all of these problems in a future release.

3.28.1.2 Deleting Directories Created by the POLYCENTER Software Installation Utility

V6.1 If a directory is created as part of a product installed using the POLYCENTER Software Installation utility and that directory is deleted manually, a reinstallation of the same product will fail. The POLYCENTER Software Installation utility determines from its database that the directory should be there, so it does not issue the command to create it. When the utility tries to reinstall the file, it fails.

Digital recommends that you not manually delete objects that were created by the POLYCENTER Software Installation utility.

3.28.1.3 Inaccurate Disk Space Reporting

V6.1 The POLYCENTER Software Installation utility estimates the amount of disk space required for an installation. Sometimes this number is inaccurate, and an installation may fail due to lack of disk space even though the utility reports that there is enough disk space.

The utility also fails to check for enabled disk quotas.

Digital expects to correct these problems in a future release.

3.28.1.4 Product Removal Restrictions

V6.1 Removing a product using the POLYCENTER Software Installation utility results in the removal of accounts created for that product. This happens regardless of whether the SYSUAF.DAT file is shared by another system disk.

The same problem exists with rights identifiers and the file RIGHTSLIST.DAT.

Digital expects to correct these problems in a future release.

3.29 POLYCENTER Striping (VAX Only)

VAX This section contains release notes pertaining to the POLYCENTER Striping product.

3.29.1 Changes and Enhancements

This section describes the replacement for POLYCENTER Striping.

3.29.1.1 StorageWorks RAID Software for OpenVMS Replaces POLYCENTER Striping

V6.2 Starting with OpenVMS Version 6.2, POLYCENTER VAX Striping is no longer supported. The supported replacement product is StorageWorks RAID Software for OpenVMS. The Striping functionality in StorageWorks RAID Software for OpenVMS is available to POLYCENTER VAX Striping customers at no charge. Please contact your Digital support representative for details of the migration plan. ♦

3.30 POSIX for OpenVMS

This section contains notes about POSIX for OpenVMS.

3.30.1 Changes and Enhancements

The following note describes a change in POSIX support.

3.30.1.1 Ordering POSIX for OpenVMS Version 3.0

V7.0 POSIX support for OpenVMS Version 7.0 is provided by POSIX for OpenVMS Version 3.0. Beginning with this version, POSIX is now available by demand order only. Contact your Digital support representative to order this product. The order number (UPI) is QA–GXXAB–H* for VAX and QA–50DAA–H* for Alpha.

3.30.2 Problems and Restrictions

The following note describes a support restriction for POSIX for OpenVMS.

3.30.2.1 POSIX for OpenVMS Version 2.0 Is Not Supported

V7.0 POSIX for OpenVMS Version 2.0 is not supported on OpenVMS Version 7.0 and later. To use the POSIX interface on OpenVMS Version 7.0, you must install POSIX for OpenVMS Version 3.0.

VAX

Caution

If you try to start up POSIX for OpenVMS Version 2.0 on an OpenVMS VAX Version 7.0 system, your system will crash. ♦

3.31 RMS Journaling

The following notes pertain to RMS Journaling for OpenVMS.

3.31.1 Problems and Restrictions

The following notes describe restrictions to RMS Journaling for OpenVMS.

3.31.1.1 Remote Access of Recovery Unit Journaled Files

V6.1

Nodes that host recovery unit journaled files that are to be accessed remotely from other nodes in the network must define SYS$NODE to be a Phase IV style node name. The node name specified by SYS$NODE must be known to any remote node attempting to access the recovery unit journaled files on the host node, and must be sufficiently unique for the remote node to use this node name to establish a DECnet connection to the host node.

3.31.1.2 VFC Format Sequential Files

VAX V5.0
Alpha V1.0

You cannot update variable fixed-length control (VFC) sequential files when using before-image or recovery unit journaling. The VFC sequential file format is indicated by the symbolic value FAB$C_VFC in the FAB$B_RFM field of the FAB.

3.32 Security Auditing

This section contains a release note pertaining to security auditing.

3.32.1 Documentation Changes and Corrections

This section describes corrections and additions to the security auditing documentation.

3.32.1.1 *OpenVMS System Manager's Manual: Tuning, Monitoring, and Complex Systems*

V7.0

The following section has been added to the version of the *OpenVMS System Manager's Manual: Tuning, Monitoring, and Complex Systems* that ships on the Version 7.0 Documentation CD–ROM. Please update your Version 6.2 hardcopy manual accordingly.

18.7.1.3 Remote Log (Archive) File

The operating system allows workstations and other users with limited management resources to duplicate their audit log files on another node. The secondary log, a security archive file, is then available to a security administrator on a remote node who has the skills to analyze the file.

Each node in a cluster must have its own archive file. An archive file cannot be shared by multiple nodes in a cluster.

Refer to Section 9.4.3.1 of the *OpenVMS Guide to System Security* for more information.

3.33 Shared Linkage Sections

This section contains notes pertaining to shared linkage sections.

3.33.1 Problems and Restrictions (Alpha Only)

V6.1

Alpha

If you want to use an alternate version of any library installed with shareable linkage, it is essential to use alternate (noninstalled) versions of all the libraries that call that library. The libraries that can be installed with shared linkage are LIBOTS, LIBRTL, CMA$TIS_SHR, DPML$SHR, and DECC$SHR.

The dependencies are in the order listed.

For example, if you issue the command:

```
$ DEFINE DPML$SHR SYS$LIBRARY:DPML$SHR.EXE;
```

then you must also issue the following command:

```
$ DEFINE LIBOTS SYS$LIBRARY:LIBOTS.EXE;
```

Failure to redefine all calling libraries may result in access violations. ♦

3.34 SORT32 Work Files

This section contains release notes pertaining to SORT32 work files.

3.34.1 Changes and Enhancements

This section describes SORT32 changes and enhancements.

3.34.1.1 Files Now Have Directory Entries

V6.2

Starting with OpenVMS Version 6.1, SORT32 work files were implemented as RMS temporary files to fix a problem wherein SORT32 work files remained if a sort was terminated abnormally. RMS temporary files do not have directory entries and are automatically deleted by RMS upon file closure. Thus, if the SORT32 image was terminated abnormally, the work files were still deleted.

Unfortunately, this implementation caused a system management restriction in that, without a directory entry, no access control information is available for the work files. Therefore, work files could be created only in a directory owned by the SORT32 user.

Typically, for large sorts, the system manager would designate a large shared temporary disk area using access control mechanisms so that the general user would require only small disk quotas. The large work files, which are deleted at the end of the sort, would be placed in the temporary area. Without a directory entry, these access control mechanisms cannot be used and the work file cannot be created in the temporary area.

Work files are now created as regular files with directory entries. Reverting to the previous behavior and allowing the management freedom for work file placement was considered more important than the occasional work file that was not deleted under abnormal conditions. Users can manually delete any work files left after a sort image terminates abnormally.

Infrequently, SORT32 might not be able to find the designated directory for the work file due to some unforeseen reason, such as a user error or hardware malfunction. In this case, SORT32 will attempt to create the work file as an RMS temporary file, possibly allowing the sort to complete successfully. This is a convenience feature and was retained for compatibility with previous versions of SORT32. Alternatively, the creation of SORT32 work files as RMS temporary files can be forced by specifying a nonexistent directory for the location of the work files.

3.35 StorageWorks RAID Array 110 Subsystem (VAX Only)

VAX

This section contains notes pertaining to the StorageWorks RAID Array 110 Subsystem.

3.35.1 Problems and Restrictions

This section describes known problems and restrictions with the StorageWorks RAID Array 110 Subsystem.

3.35.1.1 SHOW DEVICE Does Not Display Capacity

V6.1

On some VAX systems, the capacity of the StorageWorks RAID Array 110 Subsystem is not displayed after you enter the following console command:

```
>>> SHOW DEVICE
```

Instead, the capacity displays as "...". (This is due to the current settings for spin up time in the EEPROM of the StorageWorks RAID Array 110 Subsystem.)

If you reenter the SHOW DEVICE command, the correct capacity will be displayed. This will be corrected in an update to the DEC RAID OpenVMS VAX Utility Kit.

3.35.1.2 TURBOchannel Devices: Booting Not Supported

V6.1

VAX workstations running the OpenVMS VAX operating system do not provide support for system booting from a TURBOchannel device. You cannot, therefore, boot from a StorageWorks RAID Array 110 Subsystem connected to a TURBOchannel-SCSI adapter. ♦

3.36 System Management Utility

This section contains release notes pertaining to the OpenVMS System Management utility (SYSMAN).

3.36.1 Changes and Enhancements

This section describes changes and enhancements to SYSMAN.

3.36.1.1 ALF Command

V6.2

OpenVMS Version 6.2 includes changes to the following SYSMAN ALF commands:

- ADD
- REMOVE
- SHOW

The following are explanations of these changes:

ALF ADD

When you create ALF records for proxy accounts, the device parameter can be as long as 63 characters. For example:

```
SYSMAN> ALF ADD VMS:.ZKO.VMSORG.SYSMAN.CLIENT1::SYSTEM FOOBAR
```

In this command, VMS:.ZKO.VMSORG.SYSMAN.CLIENT1::SYSTEM is the value for the device parameter.

ALF REMOVE

As a result of changes in the SHOW command, you can now use REMOVE commands to remove more than one ALF record at a time. For example:

```
SYSMAN> ALF REMOVE TTA*
```

This command removes all matching records that start with the string TTA, but it does not remove any records that start with the string <nodename>$TTA, where <nodename> is the system's SCSNODE name. To remove those records, you must use one of the following commands:

```
SYSMAN> ALF REMOVE <nodename>$TTA*
```

```
SYSMAN> ALF REMOVE *TTA*
```

Note that the latter command removes all matching records that contain the string TTA in the device field. Similarly, the following command removes all matching records that have device names ending with the string TTA:

```
SYSMAN> ALF REMOVE *TTA
```

If you omit either of the wildcard characters and enter a REMOVE command, SYSMAN attempts to match the device name exactly. If more than one record matches the criteria, SYSMAN returns an error message. For example, the following command causes SYSMAN to match any record starting with <nodename>$TTA:

```
SYSMAN> ALF REMOVE TTA
```

This command produces the following error message:

```
%SYSMAN-E-ALFWILCRDREQ, more than one record might match - Wildcard or
unit number of device required.
```

Note

This message is new for OpenVMS Version 6.2. Digital recommends that you use caution when issuing REMOVE commands from Version 6.1 or lower SYSMAN clients to Version 6.2 or higher systems.

For example, the following command issued from a system running OpenVMS Version 6.1 or lower to a system running OpenVMS Version 6.2 produces no error messages and deletes all records that match FOOBAR$TTA:

```
SYSMAN> SET ENVIRONMENT/NODE=FOOBAR     ! FOOBAR runs OpenVMS Version 6.2
%SYSMAN-I-ENV, current command environment:
        Individual nodes: FOOBAR
        Username SYSTEM will be used on nonlocal nodes
SYSMAN> ALF REMOVE TTA                  ! Does not produce an error message
SYSMAN>
```

Note that if the same command is issued from a system running OpenVMS Version 6.1 or lower to another system running OpenVMS Version 6.1 or lower, it produces the following error message:

```
%SYSMAN-I-NODERR,  error returned from node FOO
-SMI-E-ALFNOMATCH, no records matched search criteria
```

This is due to incorrect processing of wildcards prior to OpenVMS Version 6.2.

ALF SHOW

The SHOW command now works as documented whether or not you use wildcards. However, there are certain restrictions in wildcard matching of ALF records. For example, the following command displays only those records that start with the string TTA:

```
SYSMAN> ALF SHOW TTA*
```

However, the following command displays only those records that start with the string <nodename>$TTA:

```
SYSMAN> ALF SHOW TTA
```

Similarly, the following command displays records that have device names ending with TTA:

```
SYSMAN> ALF SHOW *TTA
```

If you want to display all records that contain the string TTA, you must enter the following command:

```
SYSMAN> ALF SHOW *TTA*
```

3.36.1.2 Rights Identifiers

V6.2

If a SYSMAN user running with more than 125 total rights attempts to issue a SYSMAN command to a remote node within a cluster, the following error message is displayed:

```
SMI-E-RIGHTSLIM, Rights limit exceeded.
```

Note that this rights limitation includes a minimum of 3 identifiers that are granted during login when the process rights list is created:

- A UIC identifier
- A system identifier
- Depending upon the environment in which the process is operating, at least one environmental identifier

Users who want to run SYSMAN must have either:

- A separate account with no more than 125 rights
- Enough identifiers removed from their current account so that the total number of rights falls within the appropriate range

A system manager usually performs account modification or creation.

3.36.2 Problems and Restrictions

The following sections describe known problems and restrictions that apply to the System Management utility (SYSMAN).

3.36.2.1 Clusterwide DISKQUOTA Commands (Alpha Only)

V1.5

Alpha

Normally, SYSMAN DISKQUOTA commands for disks that are mounted clusterwide can be performed with the environment set to a single node. The clusterwide operation is done at the XQP level.

If the environment is set clusterwide using the /CLUSTER qualifier, the following message is displayed, and the operation still finishes correctly clusterwide:

```
%SMI-S-DQCLUS, device is mounted clusterwide, CLUSTER environment ignored
```

However, until further notice, when you use the following DISKQUOTA commands for clusterwide operation, you must first set the environment clusterwide by using the /CLUSTER qualifier:

- DISKQUOTA ENABLE/DEV=*device*
- DISKQUOTA DISABLE/DEV=*device*

If these commands are performed with the environment set to a single node, the disk quota for the clusterwide-mounted device is enabled or disabled for that node only.

The following example shows the correct usage and behavior of the DISKQUOTA ENABLE and DISKQUOTA DISABLE commands:

```
SYSMAN> SET ENV/CLUSTER
%SYSMAN-I-ENV, current command environment:
        Clusterwide on local cluster
        Username TESTUSER     will be used on nonlocal nodes

SYSMAN> DO SHOW DEV TEST01$DKA100:
%SYSMAN-I-OUTPUT, command execution on node TEST02
Device                  Device           Error    Volume         Free  Trans Mnt
 Name                   Status           Count     Label        Blocks Count Cnt
TEST01$DKA100:          Mounted              0  TESTVOL         832395     1   4
%SYSMAN-I-OUTPUT, command execution on node TEST03
Device                  Device           Error    Volume         Free  Trans Mnt
 Name                   Status           Count     Label        Blocks Count Cnt
TEST01$DKA100:          Mounted              0  TESTVOL         832395     1   4
%SYSMAN-I-OUTPUT, command execution on node TEST04
Device                  Device           Error    Volume         Free  Trans Mnt
 Name                   Status           Count     Label        Blocks Count Cnt
TEST01$DKA100:          Mounted              0  TESTVOL         832395     1   4
%SYSMAN-I-OUTPUT, command execution on node TEST01
Device                  Device           Error    Volume         Free  Trans Mnt
 Name                   Status           Count     Label        Blocks Count Cnt
TEST01$DKA100:          Mounted              0  TESTVOL         832395     1   4
```

```
SYSMAN> DISKQUOTA SHOW */DEV=TEST01$DKA100:
%SYSMAN-I-NODERR, error returned from node TEST02
-SYSTEM-F-QFNOTACT, disk quotas not enabled on this volume
%SYSMAN-I-NODERR, error returned from node TEST03
-SYSTEM-F-QFNOTACT, disk quotas not enabled on this volume
%SYSMAN-I-NODERR, error returned from node TEST04
-SYSTEM-F-QFNOTACT, disk quotas not enabled on this volume
%SYSMAN-I-NODERR, error returned from node TEST01
-SYSTEM-F-QFNOTACT, disk quotas not enabled on this volume
SYSMAN> DISKQUOTA ENABLE/DEV=TEST01$DKA100:
SYSMAN> DISKQUOTA SHOW */DEV=TEST01$DKA100:
%SMI-S-DQCLUS, device is mounted clusterwide, CLUSTER environment ignored
%SYSMAN-I-QUOTA, disk quota statistics on device TEST01$DKA100: --
Node TEST02
     UIC                  Usage        Permanent Quota   Overdraft Limit
[0,0]                     0            1000              100
[VMS,TESTUSER]            25           1000              100

SYSMAN> DISKQUOTA DISABLE/DEV=TEST01$DKA100:
SYSMAN> DISKQUOTA SHOW */DEV=TEST01$DKA100:
%SYSMAN-I-NODERR, error returned from node TEST02
-SYSTEM-F-QFNOTACT, disk quotas not enabled on this volume
%SYSMAN-I-NODERR, error returned from node TEST03
-SYSTEM-F-QFNOTACT, disk quotas not enabled on this volume
%SYSMAN-I-NODERR, error returned from node TEST04
-SYSTEM-F-QFNOTACT, disk quotas not enabled on this volume
%SYSMAN-I-NODERR, error returned from node TEST01
-SYSTEM-F-QFNOTACT, disk quotas not enabled on this volume
SYSMAN> ♦
```

3.37 System Parameters

This section contains release notes pertaining to OpenVMS system parameters.

3.37.1 Documentation Changes and Corrections

This section describes changes or corrections to the documentation of system parameters.

3.37.1.1 Changes to Online Help

V7.0 See online Help for changes or corrections to the following SYSMAN or SYSGEN system parameters:

- GBLPAGES
- GBLPAGFIL
- MAXBUF
- MAXCLASSPRI
- VIRTUALPAGECNT

In addition, the RMTDBG_SCRATCH_PAGES parameter has been renamed to DBGTK_SCRATCH.

3.38 System Time

This section contains notes pertaining to setting the system time.

3.38.1 Changes and Enhancements

The following note describes a change in system time management.

3.38.1.1 Setting Time Using the Battery-Backed Watch (BBW) (Alpha Only)

V7.0

Alpha

The OpenVMS Alpha architecture maintains the current date and time in the Battery-Backed Watch (BBW) across power failures and system downtime. The BBW is functionally equivalent to the Time of Day Register (TODR) that the VAX architecture uses. One difference, however, is the BBW's constraint on the date range.

The BBW provides sufficient storage capability for only a century. The OpenVMS Alpha system date range has been redefined as 1957 to 2056 to maintain correct leap-year processing and to provide for the millennial transition.

In addition, the OpenVMS Alpha timing mechanisms have been changed to allow 2-digit year support in the $ASCTIM system service and the DCL command SET TIME. (Prior to this change, only 4-digit year fields were allowed.) With 2-digit support, you need to enter only the last two digits of a year. The century associated with the year field is derived from the placement of the two digits in the 1957-2056 date range. For example:

```
$ SET TIME = 1-NOV-95
```

In this example, 95 is the equivalent of 1995.

```
$ SET TIME = 1-NOV-05
```

In this example, 05 is the equivalent of 2005. ♦

3.38.2 Documentation Changes and Corrections

This section describes changes and corrections in the documentation of setting system time.

3.38.2.1 *OpenVMS System Manager's Manual: Essentials*

V7.0

Section 5.8, "Setting Up Your System to Compensate for Different Time Zones," and its related subsections have been replaced with new information in the versions of this manual that ship on the Version 7.0 CD–ROM. The new information is also published in hard copy in the *OpenVMS Version 7.0 New Features Manual* in a section titled "Setting Correct Time Zone Information on Your System."

Note

The information published in the *OpenVMS Version 7.0 New Features Manual* includes a few corrections that were made after the CD–ROM versions of the *OpenVMS System Manager's Manual: Essentials* were produced. Wherever there are discrepancies between these manuals in the description about setting time, the *OpenVMS Version 7.0 New Features Manual* contains the correct information.

3.39 Terminal Fallback Facility (TFF) (Alpha Only)

V1.0

Alpha

The Terminal Fallback facility (TFF) includes a fallback driver (SYS$FBDRIVER.EXE), a shareable image (TFFSHR.EXE), a terminal fallback utility (TFU.EXE), and a fallback table library (TFF$MASTER.DAT).

Note

TFFSHR has been removed from IMAGELIB because it is not a documented, user-callable interface. The image is still available in the SYS$LIBRARY: directory.

- To start TFF, invoke the TFF startup command procedure located in SYS$MANAGER, as follows:

```
$ @SYS$MANAGER:TFF$SYSTARTUP.COM
```

- To enable fallback or to change fallback characteristics, invoke the Terminal Fallback utility (TFU), as follows:

```
$ RUN SYS$SYSTEM:TFU
TFU>
```

- To enable default fallback to the terminal, issue the following DCL command:

```
$ SET TERMINAL/FALLBACK
```

OpenVMS Alpha TFF differs from OpenVMS VAX TFF in the following ways:

- On Alpha systems, the TFF fallback driver is named SYS$FBDRIVER.EXE. On VAX systems, the TFF fallback driver is named FBDRIVER.EXE.
- *V6.1* On Alpha systems, TFF is capable of handling 16-bit character fallback. The OpenVMS Alpha fallback table library (TFF$MASTER.DAT) contains four more 16-bit character tables than the VAX library. Table 3–1 describes these additional tables.

Table 3–1 TFF Character Fallback Tables

Table Name	Base	Description
BIG5_HANYU	BIG5	BIG5 for CNS 11643 (SICGCC) terminal/printer
HANYU_BIG5	CNS	CNS 11643 (SICGCC) for BIG5 terminal/printer
HANYU_TELEX	CNS	CNS 11643 for MITAC TELEX-CODE terminal
HANGUL_DS	KS	KS for DOOSAN 200 terminal

These tables are used mainly by the Asian region. Also, the table format was changed due to the support of 16-bit character fallback.

- *V1.0* On Alpha systems, the TFU command SHOW STATISTICS does not display the size of the fallback driver (SYS$FBDRIVER.EXE).

RT terminals are not supported by TFF.

For more information about the Terminal Fallback facility, refer to the *OpenVMS Terminal Fallback Utility Manual*. ♦

3.40 UETP (User Environment Test Package)

This section contains notes pertaining to UETP.

3.40.1 Problems and Restrictions

The following note describes a support restriction for this release.

3.40.1.1 RRD45 CD–ROM Testing

V7.0

UETP does not support the RRD45 CD–ROM in OpenVMS Version 7.0. This release note describes a temporary workaround that you can use until support is added in a future release.

If you have an RRD45 CD–ROM, UETP will fail in the following manner:

```
%UETP-S-BEGIN, UETINIT01 beginning at 13-NOV-1995 16:35:22.76
 .
 .
   **********************
   *  DISK_NODE$DKA     *
   *  Error count =  1  *
   **********************
   -UETP-E-TEXT, RMS file error in file
   NODE$DKA500:NODE_NODE$DKA5000.TST
   -RMS-E-DNR, device not ready, not mounted, or unavailable
   %UETP-S-ENDED, UETDISK00 ended at 13-NOV-1995 16:36:15.10
 .
 .
   NODE$DKA:   testable     100, 200, 300
               untestable   500
 .
 .
 ***************************************************
 *                                                 *
   END OF UETP PASS 1 AT 13-NOV-1995 16:44:49.85
 *                                                 *
 ***************************************************
```

You can work around this problem and test the RRD45 device by running the CD–ROM test (UETCDRO00.EXE) against the RRD45 separately from UETP.COM. Follow these steps:

1. Edit UETSUPDEV.DAT to include the following line:

   ```
   01 36 UETCDRO00.EXE ! RRD45
   ```

2. Edit UETINIDEV.DAT to change the "N" in the row for the RRD45 device to a capital "T". The line should look like this (the number might be different):

   ```
   UCB T 00500 UETCDRO00.EXE
   ```

3. Run the CD–ROM test, for example:

   ```
   $ RUN UETCDRO00.EXE
   Controller designation?: NODE$DKA
   %UETP-S-BEGIN, UETCDRO00 beginning at 13-NOV-1995 15:15:43.20
   %UETP-I-ABORTC, CDRO_NODE$DKA to abort this test, type ^C
   %UETP-S-ENDED, UETCDRO00 ended at 13-NOV-1995 15:18:45.98
   $
   ```

3.41 VAX 7000 Systems (VAX Only)

VAX This section contains release notes pertaining to VAX 7000 systems.

3.41.1 Problems and Restrictions

This section describes problems and restrictions experienced on VAX 7000 systems.

3.41.1.1 Dump Off System Disk (DOSD) Not Supported

V6.2 Dump Off System Disk (DOSD) functionality is not supported on VAX 7000 systems. Contact your Digital support representative for a remedial kit that will enable use of Dump Off System Disk on VAX 7000 systems. ♦

3.42 VAX System Dump Analyzer (SDA) (VAX Only)

VAX The following notes pertain to the VAX System Dump Analyzer (SDA).

3.42.1 Changes and Enhancements

The following note describes a change to VAX SDA.

3.42.1.1 Dump File Process

V7.0 OpenVMS VAX SDA now uses RMS file access to process the dump file instead of mapping the dump file into the working set. Because of this change, some commands that execute over a large range, such as SEARCH, may take longer.

Another effect of this change is that a value of 16,000 for the system parameter VIRTUALPAGECNT should be sufficient to analyze any dump, even if a large number of symbols is read in.

3.42.2 Problems and Restrictions

This section describes known System Dump Analyzer problems and restrictions.

3.42.2.1 EXAMINE Command

V6.0 If you make a mistake specifying a virtual address for the EXAMINE command and you are examining global page table entries, your system may crash with a bugcheck. This occurs rarely and only when you use ANALYZE/SYSTEM. ♦

3.43 VMScluster Systems

The following sections contain information pertaining to VMScluster systems.

3.43.1 Changes and Enhancements

This section contains notes about changes to VMScluster systems.

3.43.1.1 SCSI VMScluster Systems

V6.2 Starting with OpenVMS Version 6.2, VMScluster systems support the Small Computer Systems Interface (SCSI) as a storage interconnect. See the *OpenVMS Version 7.0 New Features Manual* for a description of this feature, including configuration information, installation information, and a description of any known restrictions and problems.

3.43.1.2 Concurrency Improvements

V6.1

OpenVMS Version 6.1 and later provides an improved file system. This improved file system is also available as a special release known as the XQP+ for PATHWORKS.

One of the features provided by the improved file system is improved concurrency, which allows multiple processes to create files on a single disk in parallel.

In a VMScluster system, do not enable improved concurrency unless all nodes are running the improved file system. Each node in the cluster must be running one of the following:

- OpenVMS Alpha Version 6.1 or later
- OpenVMS VAX Version 6.1 or later
- XQP+ for PATHWORKS

If you are running the improved file system on all the nodes in your VMScluster, turn on improved concurrency by setting the static XQPCTL2 system parameter to 1.

3.43.1.3 FDDI Clusters (Alpha Only)

V6.1

Alpha

Beginning with OpenVMS Alpha Version 6.1, you no longer have to set system parameter PE3 to 1 to enable FDDI clustering. In fact, system parameter PE3 should not be set to any value other than 0. ♦

3.43.2 Problems and Restrictions

This section describes problems and restrictions pertaining to VMScluster systems.

3.43.2.1 VMScluster Tape Serving Requires Adding TMSCP Parameter to MODPARAMS.DAT

V7.0

To enable tape serving in a VMScluster, you can run the CLUSTER_CONFIG.COM or CLUSTER_CONFIG_LAN.COM command procedure and do the following:

1. Select `CHANGE a cluster member's characteristics`.
2. From the CHANGE menu, select `Enable` *nodename* `as a tape server`.
3. Respond to the prompt `Enter a value for` *nodename's* `TAPE_ALLOCLASS parameter [0]:`

Before or after you enable tape serving, you must also add the TMSCP_SERVE_ALL=*n* system parameter to SYS$SYSTEM:MODPARAMS.DAT.

The values for *n* in TMSCP_SERVE_ALL=*n* follow:

- 0=Do not serve tapes (the default value)
- 1=Serve all available tapes (local and remote)
- 2=Serve only locally connected tapes

You cannot rely on AUTOGEN or SYSGEN to enable tape serving. The SYSGEN default for this system parameter is no tape serving, and AUTOGEN does not override this default, even after you enabled tape serving with CLUSTER_CONFIG.COM or CLUSTER_CONFIG_LAN.COM.

This extra step of adding the TMSCP_SERVE_ALL=*n* system parameter to SYS$SYSTEM:MODPARAMS.DAT will be eliminated in the next version of OpenVMS.

3.43.2.2 KFESA/KFESB Controller Restrictions

V6.2

There are restrictions in the use of the KFESA/KFESB EISA DSSI controllers in systems that are members of a VMScluster system. This configuration will be supported only after a specific update is applied to your systems. This update may be present on your OpenVMS V6.2 CD–ROM distribution kit in the following location:

[KFESX]KFESX_UPDATE.BCK

To apply this update, restore the save set to a scratch directory and follow the instructions in the READ-ME.1ST file.

If this update save set is not present on your CD–ROM distribution kit, please contact your Digital support representative and ask for the KFESA/KFESB TIMA kit.

3.43.2.3 System Startup in a VMScluster Environment (Alpha Only)

V6.2

Alpha

In a VMScluster environment, under some circumstances the system startup procedure may fail to write a new copy of the ALPHAVMSSYS.PAR file. If this occurs, the console output from the boot sequence reports the following messages:

```
%SYSGEN-E-CREPARFIL, unable to create parameter file
-RMS-E-FLK, file currently locked by another user
```

This error creates an operational problem only when changing SYSGEN parameters using a conversational boot. For a normal, nonconversational boot, this error message is purely cosmetic because the parameter file has not changed. If a conversational boot is used, and SYSGEN parameters are changed at boot time, these changed parameters will be correctly used for the current boot of the system. However, since the boot procedure does not successfully write a new copy of the parameter file, these changed parameters will not be used in subsequent boots.

To permanently change SYSGEN parameters that have been changed by a conversational boot, run SYSGEN after the system has completed booting, and execute the following commands:

```
SYSGEN> USE ACTIVE
SYSGEN> WRITE CURRENT ♦
```

3.44 Volume Shadowing

The following sections pertain to volume shadowing software.

3.44.1 Changes and Enhancements

This section describes changes to volume shadowing software.

3.44.1.1 Timer-Based Polling

V6.2

Shadowing now uses a timer to adhere more accurately to the number of seconds specified by the SHADOW_MBR_TMO parameter. For directly connected SCSI devices that have been powered down or do not answer to polling, the elapsed time before a device is removed from a shadow set can still approach 1 minute. In all other situations, the elapsed time will closely approximate the number of seconds specified by the SHADOW_MBR_TMO parameter.

3.44.1.2 Volume Shadowing Locally Connected SCSI Disks (VAX Only)

V6.2

VAX

The VAX SCSI disk driver (DKDRIVER) does not implement the same level of error handling that exists in all other OpenVMS disk drivers. Consequently, the OpenVMS Volume Shadowing product is unable to recover from several rare error conditions when used on SCSI disks that are locally connected to VAX 3000 and VAX 4000 series systems. This problem does not occur on CI-based or DSSI-based SCSI storage.

Problems resulting from the unimplemented error handling typically are shadow set hangs or system crashes. No evidence of data corruption has been seen in Digital test labs. No hangs or crashes at customer sites have been reported. The complexity of this problem, which has existed since Version 5.5-2, precludes it being fixed with a patch, and Digital has no plans to implement the missing error handling code.

Customers who wish to use OpenVMS Volume Shadowing on SCSI disks that are locally connected to VAX 3000 and 4000 series systems will be supported by Digital up to the limits of the current error handling capabilities. Volume Shadowing will continue to deliver data availability in the event of disk media failure on these systems, although some errors will cause a system failure, necessitating a reboot.

Digital recommends that customers upgrade to the latest version of Volume Shadowing for OpenVMS in order to benefit from the latest patches. Digital has attempted to minimize the impact of this error handling flaw, and the latest version of Volume Shadowing recovers significantly better than earlier versions. ♦

3.44.2 Problems and Restrictions

The following sections describe known problems and other considerations pertaining to volume shadowing.

3.44.2.1 Failure to Write Crash Dump after Removal of Boot Member (Alpha Only)

V7.0

Alpha

If a boot device is removed from a multiple-member system disk shadow set, OPCOM issues the following messages, warning you that this removal will result in the loss of a crash dump, should a subsequent system crash occur:

```
DSA18: shadow master changed.  Dump will NOT be written if system crashes.

$4$DUA18: (RED90A, CSG5) has been removed from shadow set.

DSA18: shadow set has been reduced.
```

If a crash occurs any time after this, the following message is displayed, indicating that the crash dump cannot be written:

```
****  Unable to take system dump, master member of
****  system disk shadow set wrong, Code = 00008001
```

In ordinary circumstances, crash dump information is written to the boot device, which is usually the master member of the system disk shadow set.

However, if the boot device is removed from the shadow set, shadowing will do the following:

1. Select another member of the shadow set to become the master member.
2. Disable the system crash dump. This step is taken to prevent the primitive drivers from writing to that device at bugcheck time, since the device is no longer a part of the system disk shadow set.

When the boot device is returned to the system disk shadow set, it becomes the target of a full copy operation.

Even after the copy operation has completed and the returned device becomes a full member, a crash dump still cannot be written. This is because the master member designation is not moved back to the boot member, resulting in failure to write a crash dump if a subsequent system crash occurs.

Although not a desirable availability solution, a workaround is to wait for the boot device to complete the copy operation, and then remove *all* other members of the shadow set. This moves the master member designation to the boot member and, in the event of a crash, enables a dump to be written to the boot member.

Digital plans to address this problem in the next release and provide remedial fixes back to OpenVMS Alpha Version 6.1. ♦

3.44.2.2 SHADZEROMBR Bugcheck for Synchronization Problem

V6.2 A synchronization problem which resulted in a virtual unit having zero members formerly caused ACCVIO system failures in Volume Shadowing for OpenVMS. Because having zero members is an illegal state for a mounted virtual unit, Volume Shadowing crashed the node in an attempt to maintain data integrity.

Version 6.2 of OpenVMS provides a new shadowing bugcheck called SHADZEROMBR. If this bugcheck occurs, please document and report what events occurred on other nodes in the VMScluster environment that also have the shadow set mounted. For example, if Node A and Node B share the same shadow set, report any problems that occurred on Node A before Node B failed with a SHADZEROMBR error. This will help OpenVMS Engineering determine what caused the failure.

To continue operation, reboot the failed node and remount the shadow sets.

3.44.2.3 KDM70 Devices

V6.1 Volume Shadowing for OpenVMS (phase II) requires that KDM70 disk controllers run a minimum of Version 3.0 microcode.

3.44.2.4 DISMOUNT/CLUSTER Command Can Cause Clusterwide Hang

V6.1

When you issue the DISMOUNT/CLUSTER command to a shadow set, it can sometimes cause a clusterwide hang. If you experience this problem, use the DISMOUNT/SYSTEM command for every node in the VMScluster. You can enter this command once for every node in the VMScluster by using the SYSMAN DO command in combination with the DISMOUNT/SYSTEM command.

For more information about using the SYSMAN DO command, see the *OpenVMS System Manager's Manual*.

3.44.2.5 RZ57 Support Restricted in Shadow Sets

V6.1

OpenVMS systems support the RZ57 disk drive only with device revision level D01 and microcode revision 6000 or higher.

Disks that meet these conditions can be shadowed.

3.44.3 Documentation Changes and Corrections

This section contains a correction to the volume shadowing documentation.

3.44.3.1 *Volume Shadowing for OpenVMS* Manual

V6.1

The first bullet in Section 3.1 of the *Volume Shadowing for OpenVMS* manual incorrectly indicates that only Alpha systems support per-disk volume shadowing licensing. Starting with Version 6.1, OpenVMS VAX systems also support per-disk licensing.

4 Programming Release Notes

This chapter provides release notes about both application and system programming on OpenVMS systems.

For information about new programming features included in OpenVMS Version 7.0, see the *OpenVMS Version 7.0 New Features Manual*.

Alpha

Note About Privileged-Code Interfaces

OpenVMS Alpha Version 7.0 includes significant changes to OpenVMS Alpha privileged interfaces and data structures. As a result of these changes, some customer-written device drivers and applications with inner-mode code may require source changes.

For more details about changes to OpenVMS Alpha privileged interfaces and data structures, see the *OpenVMS Alpha Guide to Upgrading Privileged-Code Applications*. ♦

4.1 Batch and Print Queues

This section contains release notes pertaining to batch and print queues.

4.1.1 Problems and Restrictions

This section describes problems and restrictions pertaining to batch and print queues. For problems important to system managers, refer to Section 3.5.2.

4.1.1.1 Terminating Executing Batch Jobs

V6.2 Under the following conditions, the DELETE/ENTRY command might fail to stop an executing batch job:

- The batch job is a DCL command procedure.
- There is an ON ERROR CONTINUE command (or SET NOON command) within the command procedure.

The DELETE/ENTRY command causes the job to terminate in phases. A delete_process AST routine is given in user mode, supervisor mode, and then executive mode. Because there is a small delay between each mode, it is possible that, in a batch job, a user-mode image might terminate and the command procedure might continue to execute until the supervisor-mode delete_process AST routine is executed.

The return status of the SYNCHRONIZE command is assumed to contain the termination status of the target batch job. In addition, command procedures would normally execute a command such as $ON ERROR THEN CONTINUE or $SET NOON before issuing the SYNCHRONIZE command. If a DELETE /ENTRY command is issued to the job executing the SYNCHRONIZE command, the JBC$_JOBABORT is interpreted as being the termination status of the target batch job rather than a return status of the SYNCHRONIZE command. The command procedure then continues to execute for a short period with this incorrect assumption and performs an operation such as requeuing the target batch job or incorrectly reporting a failure of the target batch job.

This problem has been fixed for the SYNCHRONIZE command by detecting this situation and waiting in an exit handler for longer than the delay between the user and supervisor mode termination delay.

Any other images that would report the job completion status obtained by the SJC$_SYNCHRONIZE_JOB function code of the $SNDJBC system service as the return status of the program should implement logic similar to the following:

1. Declare an exit handler
2. In the exit handler, implement the following logic:

```
IF (exit status is JBC$_JOBABORT)
THEN
    Wait 10 seconds
ENDIF
```

4.2 Debugger

This section contains release notes pertaining to the OpenVMS Debugger. Debugger release notes specifically about system management are in Section 3.8.

Unless specified otherwise, the release notes apply to both the character-cell and DECwindows Motif interfaces of the debugger.

4.2.1 Changes and Enhancements

This section describes changes in OpenVMS Debugger Version 7.0.

4.2.1.1 Depositing Values in Arrays: New Error Message

V7.0

If a programmer deposits a value past the end of an array (a common programming error), he or she may inadvertently corrupt the invocation context register set and stack, causing latent problems to occur further down the program. When the debugger discovers that the invocation context is corrupted, the debugger displays an appropriate error message, directing the user toward the program for the cause of corruption. (Previously, the library routines responsible for manipulating invocation context sometimes failed to update the context.)

4.2.2 Problems and Restrictions

This section describes known debugger problems and restrictions.

4.2.2.1 COBOL INITIALIZE Statement and Large Tables (Arrays) (Alpha Only)

V7.0

Alpha

On OpenVMS Alpha systems, the debugger can take an unusually great amount of time and resources if you use the STEP command to execute an INITIALIZE statement in a COBOL program when a large table (array) is being initialized.

In order to work around this problem, set a breakpoint on the first executable line past the INITIALIZE statement, rather than stepping across the INITIALIZE statement. ♦

4.2.2.2 EXAMINE Command with Large Arrays

V7.0

When you attempt to examine a very large array with the EXAMINE command, the attempt may fail, with memory exhaustion and the %DEBUG-E-NOFREE error message.

The workaround is to examine the array in increments to release dynamic memory after each command. Consider a 400-slice array (table) initialized as follows in this fragment of a COBOL program:

```
IDENTIFICATION DIVISION.
PROGRAM-ID.  TRIAL.
ENVIRONMENT DIVISION.
DATA DIVISION.
WORKING-STORAGE SECTION.
01 RANK_WEEK_ARRAYS.
   02 RANK_PART OCCURS 400 TIMES.
      03 WEEK_PART OCCURS 50 TIMES.
         05 CLE_SCREENS     PIC 9(3).
         05 CLE_FSS         PIC 9(9).
         .
         .
         .
PROCEDURE DIVISION.
001-BEGIN.
    INITIALIZE RANK_WEEK_ARRAYS.
```

In the debugger, you can examine selected slices of the array as follows:

```
DBG> EXAMINE WEEK_PART(1,1:50)
DBG> EXAMINE WEEK_PART(2,1:50)
DBG> EXAMINE WEEK_PART(3,1:50)
.
.
.
```

To examine the entire large array, use a simple debugger command procedure like the following:

```
$ CREATE DBG.COM
DEFINE/VALUE INDEX=1
WHILE (INDEX <= 400) -
  DO ( EXAMINE WEEK_PART(INDEX,1:50); -
       DEFINE/VALUE INDEX=INDEX+1 )
DELETE INDEX Ctrl/Z
```

Execute this from within the debugger as follows:

```
DBG> @DBG.COM
```

The DEFINE/VALUE command defines a local debugger variable that is used within the command procedure to keep track of the outer array index (RANK_PART) of the array, RANK_WEEK_ARRAYS. The command procedure then uses the index to loop through and examine each of the 400 slices of the array. When finished examining the entire array, the command procedure cleans up after itself by deleting the local debugger variable.

4.2.2.3 Heap Analyzer and /NODEBUG (Alpha Only)

V7.0

Alpha

On OpenVMS Alpha systems, the Heap Analyzer does not work on programs linked with the /NODEBUG qualifier. ♦

4.2.2.4 DEBUG.EXE Image (VAX Only)

V6.1

VAX

On OpenVMS VAX systems, you cannot debug an image named DEBUG.EXE. This name is used internally for the kernel portion of the debugger. To work around this problem, do the following:

```
$ DEFINE LIB$DEBUG SYS$SHARE:DEBUG
```

Optionally, you can rename your image to something other than DEBUG.EXE. ♦

4.2.2.5 Watchpoint Support

V7.0

The following are problems and restrictions with the debugger's support for watchpoints:

- An active static watchpoint can cause a system service to fail, likely with an ACCVIO status, if the system service is not supported by the system service interception (SSI) vehicle (DBGSSISHR on OpenVMS VAX systems and SYS$SSISHR on OpenVMS Alpha systems). Any system service that is not in SYS$PUBLIC_VECTORS is unsupported by SSI, including User Written System Services (UWSS) and any loadable system services.

  When a static watchpoint is active, the debugger write-protects the page containing the variable to be watched. A system service call not supported by SSI can fail if it tries to write to user memory.

  The workaround is to do either of the following:

  - Deactivate the static watchpoint before the service call, and check it manually and reactivate it after the call completes.
  - Use nonstatic watchpoints. Note that nonstatic watchpoints can slow execution. See the *OpenVMS Debugger Manual* for a discussion of static and nonstatic variables and the use of the SET WATCH/[NO]STATIC command.

- An active static watchpoint can cause changes in program behavior when both of the following conditions exist:

  - The program makes a system service call that is supported by SSI.
  - As a result of that system call, the program branches based on information received about whether or not ASTs or thread switching are enabled.

  When a static watchpoint is active, the debugger write-protects the page containing the variable to be watched. Before the program being debugged makes the system service call, the debugger uses SSI to deactivate the watchpoint, disable ASTs, and disable thread switching. Immediately after the system service call, the debugger restores the state of the watchpoint, ASTs, and thread switching. This can alter the behavior of a program being debugged that makes a system service call if the call itself depends on or returns information based on the state of AST or thread switching enabling.

  As with system service calls not supported by SSI, the workaround is to do either of the following:

  - Deactivate the static watchpoint before the service call, and check it manually and reactivate it after the call completes.

  - Use nonstatic watchpoints. Note that nonstatic watchpoints can slow execution. See the *OpenVMS Debugger Manual* for a discussion of static and nonstatic variables and the use of the SET WATCH/[NO]STATIC command.

VAX V6.0
Alpha V1.0

- Watchpoints set on variables whose addresses are in global sections do not work. Attempting to set a watchpoint on a location in a global section results in a %DEBUG-E-BADWATCH message.
- If a watched location changes during a system service routine, you will be notified, as usual, that the watchpoint occurred. However, the stack will show one or more debugger frames on top of the frame or frames for your program.

  To work around this problem, enter one or more STEP/RETURN commands to get back to your program.

4.2.2.6 SHOW CALLS Command (Alpha Only)

V1.5

Alpha

On OpenVMS Alpha systems, after you enter a SHOW CALLS command, the output may include system call frames in addition to the user call frames associated with your program. System call frames appear in the following circumstances:

- When an exception occurs
- When an asynchronous system trap occurs
- When a watchpoint occurs in system space

The display of system call frames does not indicate a problem. ♦

4.2.2.7 DECwindows Motif Interface

V7.0

The following problems or restrictions are specific to the DECwindows Motif interface:

- In the DECwindows Motif interface, when you enter an EDIT command at the command prompt, the debugger utilizes the DECterm window that invoked the debug session as the user-defined-editor window (as opposed to the debugger's built-in editor, which is hardwired to the COMMANDS EDIT FILE pull-down menu). This behavior constitutes a tradeoff that allows a more flexible choice of editors. If you inadvertently exit this DECterm window using FILE EXIT or MWM Close, the debug session terminates abruptly, having lost its parent window.
- You cannot use the DECwindows Motif interface to the debugger to debug detached processes such as print symbionts that run without a command line interpreter (CLI). The design and implementation of the debugger's DECwindows Motif interface requires that the process being debugged have a CLI. To debug a detached process that does not have a CLI, you must use the character-cell (screen mode) interface to the debugger.

  To do so, define DBG$INPUT and DBG$OUTPUT to be directed to a terminal port that is not logged in. This allows the image to be debugged with the standard character-cell interface on that terminal.

  The following example assumes that you are debugging a print symbiont that was compiled and linked using the /DEBUG qualifiers:

```
$ DEFINE/GROUP DBG$INPUT  TTA3:
$ DEFINE/GROUP DBG$OUTPUT TTA3:
$ DEFINE/GROUP DBG$ERROR  TTA3:
$ START/QUEUE SYS$PRINT /PROCESSOR=dev:[dir]test_program
```

```
[Debugger starts up on logged-out terminal TTA3:]
```

VAX V6.0
Alpha V1.5

- Occasionally, if you are debugging a UI application and you have many debugger windows overlapping the user program's windows, the X server will abruptly terminate the user program.

  To avoid this problem, refrain from overlapping or covering windows belonging to the user program.

- If you are stopped at a breakpoint in a routine that has control of the mouse pointer by a PointerGrab or a KeyboardGrab, your workstation will hang.

  To work around this problem, debug your program using two workstations. For more information, see the *OpenVMS Debugger Manual*.

- Table 4–1 lists debugger commands that are not available in the DECwindows Motif interface. The debugger issues an error message when it encounters one of these commands at the command prompt or in a command procedure.

Table 4–1 Debugger Commands Not Available in the DECwindows Motif Interface

ATTACH	SELECT
CANCEL MODE	(SET, SHOW) ABORT_KEY
CANCEL WINDOW	(SET, SHOW) KEY
DEFINE/KEY	(SET, SHOW) MARGINS
DELETE/KEY	SET MODE [NO]KEYPAD
DISPLAY	SET MODE [NO]SCREEN
EXAMINE/SOURCE	SET MODE [NO]SCROLL
EXPAND	SET OUTPUT [NO]TERMINAL
EXTRACT	(SET, SHOW) TERMINAL
HELP	(SET, SHOW) WINDOW
MOVE	(SHOW, CANCEL) DISPLAY
SAVE	SHOW SELECT
SCROLL	SPAWN

- DECwindows Motif does not provide for specialized key support (such as Ctrl/Y), but the DECwindows Motif interface provides alternative means of executing these functions (for example, the STOP button).

- If you have not installed a DECwindows Motif license (DW-MOTIF) on the node from which you are using the debugger, the debugger's GUI fails with the following error messages:

  ```
  %NONAME-W-NOMSG, Message number 00000000
  -DEBUG-F-FATALSTATUS, a fatal condition was detected by the debugger.
  %DEBUG-F-INITERR, an error has occurred during debugger initialization,
  unable to continue this session.
  ```

  To return to DCL level after this failure, enter a Ctrl/C or Ctrl/Y key sequence. To correct the problem, install a DECwindows Motif license or use the character-cell interface by defining DBG$DECW$DISPLAY to " ".

- Occasionally, the DECwindows Motif interface does not accept properly formatted Fortran commands. For example, if you select the fragment "qdata(:ilen)" from the program statement "read(qdata(:ilen),100)ivalue" with the mouse and paste it into an EXAMINE command at the DBG> prompt, the debugger issues an error message:

  ```
  DBG>  EXAMINE qdata( :ilen)
  DEBUG-E-MISOPEMIS, misplaced operator or missing operand at 'end of
  expression'
  ```

  To work around this problem, reenter your command:

  ```
  DBG>  EXAMINE qdata(1:ilen)
  ```

4.2.2.8 EXAMINE LABEL[*n*] Command (Alpha Only)

V1.5

Alpha

On OpenVMS Alpha systems, a command in the form EXAMINE LABEL[*n*] or EXAMINE LABEL(*n*), where LABEL is a label for a code location and *n* is an integer, causes an access violation error. In this case, the debugger does not handle the error.

Note that this problem does not occur when the label marks the start of data storage, as in a MACRO program. ♦

4.2.2.9 Kept Debugger

VAX V6.0
Alpha V1.5

The following problems or restrictions are specific to the Kept Debugger:

- If a previous debugger process has not completely stopped, you may see the following error at debugger startup:

  ```
  %DEBUG-E-INTERR, internal debugger error in
     DBGMRPC\DBG$WAIT_FOR_EVENT got an ACK
  ```

  To fix this problem, exit the debugger. Then use the DCL command SHOW PROCESS/SUBPROCESS to check if any debugger subprocesses exist. If so, you can stop them by using the DCL command STOP. You should then be able to restart the debugger without seeing the error described above.

- Running a sequence of many large programs may cause the debugger to fail because it has run out of memory, global sections, or some other resource.

  To fix this problem, exit the debugger and restart the debugging session.

- Many commands are disabled when there is no running program. This includes commands that might be expected to work, such as SET STEP and SET PROMPT/SUFFIX. The disabled command may cause DBG$INIT files to generate %DEBUG-W-NOPROGRAM messages. These commands are enabled once a RUN command has been executed.

- The prompt may change when a RUN command is executed. It will change back to its former state once the program has completed.

- If you are using the DECwindows Motif interface (as opposed to character-cell screen mode), and you try to run a program that does not exist, or misspell the name of a program that does exist, the following error messages are displayed in the DECterm window, rather than in the command view:

  ```
  %DCL-W-ACTIMAGE, error activating image
  -CLI-E-IMAGEFNF, image file not found
  ```

  To avoid this problem, make sure the “Select an application to run” box in the File Selection popup contains a valid file specification.

- The %DEBUG-I-INITIAL is not displayed after execution of the RERUN /SAVE command. The absence of this message does not adversely affect the execution of this command.
- If you enter the RERUN command while your file (that is, the image you wish to rerun) is locked by another user, the debugger returns the following message:

  ```
  %DEBUG-E-NORERUNPGM, There is no program to RERUN
  ```

  This situation might occur, for example, if another user has changed the protection on the file. To work around this problem, have your system manager change the protection. Then, enter a RUN command and reset breakpoints.

4.2.2.10 Complex Variables in Fortran Programs (Alpha Only)

V1.0

Alpha

On OpenVMS Alpha systems, the debugger cannot evaluate expressions that contain complex variables. (Currently, DEC Fortran is the only supported language that provides complex variables.)

To work around this problem, examine the complex variable and then evaluate the expression using the real and imaginary parts of the complex variable as shown by the EXAMINE command. ♦

4.2.2.11 Inlined Routines

VAX V7.0
Alpha V1.0

On OpenVMS systems, the debugger does not support attempts to debug inlined routines. If you attempt to debug an inlined routine, the debugger issues a message that it cannot access the routine, as shown in the following example:

```
%DEBUG-E-ACCESSR, no read access to address 00000000
```

To work around this problem, compile your program with the /NOOPTIMIZE qualifier.

4.2.2.12 STEP/INTO Command and User Exception Handlers (Alpha Only)

V1.0

Alpha

On OpenVMS Alpha systems, when execution is stopped at an exception break, STEP/INTO does not transfer control to a user exception handler. To work around this problem, set a breakpoint on the handler. ♦

4.2.2.13 STEP/OVER Command Error with One-Line Program Loops (Alpha Only)

V1.0

Alpha

On OpenVMS Alpha systems, when you issue the STEP/OVER command at a program loop that is coded on a single source line, and that source line also contains a routine call, the debugger steps into the called routine instead of stepping to the next source line. In the following example, if you issue the STEP /OVER command when execution is stopped at the FOR loop, the debugger steps into the square routine instead of stepping to the j assignment statement:

```
for (i=0;i<10;i++) square(i);
j=6;
```

To work around this problem, either set a temporary breakpoint on the line following the FOR loop (in the previous example, j=6), or move the routine call to a separate line, as follows:

```
for (i=0;i<10;i++)
   square(i); ♦
```

4.2.2.14 Vector Support (VAX Only)

V5.4

VAX

On OpenVMS VAX systems, the following are problems and restrictions with the debugger's support for vectorized programs:

- When the programming language is BLISS, COBOL, or RPG, you must specify a type qualifier to deposit into %VMR. For example:

  ```
  DEPOSIT/QUADWORD %VMR = %HEX 0FFFFFFFF
  ```

- When the programming language is PL/I, COBOL, or DIBOL, the command EXAMINE %VMR displays %VMR as an array of bits instead of as a hexadecimal quadword. Enter the command EXAMINE/HEX/QUADWORD %VMR to obtain the default behavior for other programming languages.
- When the vector mode is synchronized (that is, if you have entered the command SET VECTOR_MODE SYNCHRONIZED), the debugger suspends execution twice at any breakpoints that were set on vector instructions. To resume execution from such breakpoints, you must enter the GO or STEP command twice. ♦

4.2.2.15 Improper Display of Long Text Strings

V7.0

The debugger improperly displays long text strings in screen mode, sometimes wrapping the display in inconvenient places. This problem will be corrected in a future release.

To work around this problem, use the SCROLL/LEFT and SCROLL/RIGHT commands or keypad functions KP4 (SCROLL/LEFT) and KP6 (SCROLL/RIGHT).

4.2.3 Corrections

This section describes corrections to problems that existed in OpenVMS Debugger Version 6.2.

4.2.3.1 Ada Boolean Expressions

V7.0

Previously, an attempt to evaluate a Boolean expression in an Ada program sometimes resulted in a wrong answer. This problem has been corrected.

4.2.3.2 Ada Enumeration Variables

V7.0

Previously, the debugger had some difficulty in handling Ada names and the relationship between Ada package specifications and package bodies. One of the symptoms of this problem was that the debugger produced inconsistent results—sometimes issuing the OPTNOTALLOW error message instead of the NOUNIQUE error message—when two or more Ada enumeration types contained the same enumeration value. This problem has been corrected.

4.2.3.3 Arrays: Variable Used as Index

V7.0

Previously, if a variable was used as an array index in a debugger command (EXAMINE, for example), the debugger sometimes issued an OPTNOTALLOW error message. This problem has been corrected.

4.2.3.4 COBOL INITIALIZE Statement and Large Tables (Arrays) (VAX Only)

V7.0

VAX

In past versions of the OpenVMS operating system, the debugger could take an unusually great amount of time and resources to step past an INITIALIZE statement in a COBOL program when a large table (array) was being initialized. This problem has been corrected on VAX; it still exists on Alpha. ♦

4.2.3.5 COBOL Module with the Same Name as a Variable

V7.0

Previously, if you set a breakpoint on a COBOL module that had the same name as a variable, the debugger did not stop on the module but let the program run to completion, because the breakpoint was set on the variable, rather than the beginning of the module. The problem has been corrected, and now the debugger properly issues the DBG$_BPTONDATA message ("execution point or tracepoint set on data item").

4.2.3.6 COBOL Scaled Computational Data (Alpha Only)

V7.0

Alpha

Previously, on OpenVMS Alpha systems, when COBOL scaled computational data was examined, the decimal point did not appear and the number was displayed as a whole number. This problem has been corrected. ♦

4.2.3.7 Customizations of Windows Interface

V7.0

Previously, in the DECwindows Motif interface, user customizations were not properly saved or restored, and users were unable to view saved customizations in the debugger editor. This problem has been corrected.

4.2.3.8 %DEBUG-E-LINEINFO Message Information on Line Numbers

V7.0

Previously, on Version 6.0 or higher, the debugger sometimes failed to return the next line number or the previous line number in a %DEBUG-E-LINEINFO ("no line nnn") message. This message is issued when you try to set a breakpoint (SET BREAK %line nnn) or examine a source line (EXAMINE/SOURCE %line nnn) and the line number is outside a procedure. This problem has been corrected, and the message now includes the information.

4.2.3.9 DEPOSIT Command with Large Unsigned Integer

V7.0

Previously, an attempt to deposit a decimal unsigned longword integer greater than the maximum signed longword (2147483647) resulted in the %DEBUG-W-UNACVT message ("unable to convert decimal...to longword logical"). The problem has been corrected, and deposits of this type are now successful.

4.2.3.10 EVALUATE Command and Truncation Problem (Alpha Only)

V7.0

Alpha

Previously, on OpenVMS Alpha systems, when you issued the EVALUATE command for an integer, the debugger truncated the return value if it was larger than a longword. This problem has been corrected. ♦

4.2.3.11 Fortran Complex Numbers (Alpha Only)

V7.0

Alpha

Previously, on OpenVMS Alpha systems, an attempt to examine complex numbers sometimes incorrectly resulted in an "illegal floating point number" message. This problem has been corrected. ♦

4.2.3.12 Heap Analyzer and Memory

V7.0

Previously, for some applications that did many get and free operations at asynchronous system trap (AST) level, the Heap Analyzer's kernel zone could grow very large, causing the application to use so much memory that the Heap Analyzer became unusable. This problem has been corrected.

4.2.3.13 Library Text Modules and SEARCH Command

V7.0

Previously, a debugger SEARCH command sometimes resulted in a UNAOPNSRC ("unable to open source file") message when header files from a text library were included. This problem has been corrected.

4.2.3.14 Null Frame Procedures and SHOW CALLS (Alpha Only)

V7.0

Alpha

Previously, on OpenVMS Alpha systems, a problem with null frame procedures prevented the SHOW CALLS command from displaying full symbolic and line number information about null frame routines in the SHOW CALLS display. This problem has been corrected. For more information on null frame procedures, see the *OpenVMS Calling Standard*. ♦

4.2.3.15 SET WATCH, STEP, and GO Commands (Alpha Only)

V7.0

Alpha

Previously, on OpenVMS Alpha systems, a SET WATCH command could cause an ACCVIO status in the debugger if the SET WATCH command was nonstatic and on a large union that caused the kernel to expand its memory. Also, if you issued a SET WATCH command and subsequently issued a STEP or GO command, an OPCDEC condition could result. This problem has been corrected. ♦

4.2.3.16 Incorrect Source Code Displayed

V7.0

Previously, when the debugger got control and tried to display source code corresponding to the current location, it was not always successful when it did not have the source code for the PC at the top of the stack. This problem has been corrected.

4.2.3.17 Incomplete Program State Information

V7.0

Previously, when large amounts of data were involved, the debugger would sometimes provide incomplete information about the state of the program being debugged and display a %DEBUG-E-PARTIALINFO ("There was an error in DEBUG's RPC...") message. This problem has been corrected.

4.2.3.18 Inaccurate Displays in Multiprocess Applications

V7.0

Previously, during debugging of a multiprocess application in which the processes share a global data section, deposits to the shared global data in one process were not always visible in the other processes. This problem has been corrected.

4.3 DEC Ada Run-Time Library

This section contains release notes pertaining to the DEC Ada Run-Time Library.

4.3.1 Problems and Restrictions

This section describes a potential problem.

4.3.1.1 Unexpected Storage Errors (Alpha Only)

V7.0

Alpha

In OpenVMS Alpha Version 7.0, binary compatibility fails for some DEC Ada programs that make incorrect assumptions about the amount of task space used by DEC Ada tasks. If a task gets a storage error that it did not previously get, you may need to add a length clause specifying the storage size for the task. If you already use a length clause, you might need to increase the amount of storage specified. This is necessary only in cases where the specified size (or default size) is not large enough for the task's execution. ♦

4.4 DEC C++

The following sections contain release notes pertaining to DEC C++.

4.4.1 Problems and Restrictions

This section describes known DEC C++ problems and restrictions.

4.4.1.1 SYS$LIB_C.TLB Structures Need Prototypes (Alpha Only)

V6.1

Alpha

DEC C++ users who include header files from SYS$LIB_C.TLB may receive compiler errors. Many of the structures within SYS$LIB_C.TLB contain pointers to other structures. These pointers are defined using the tag name of the structure to which it points. In C, if you declare something using a tag name, it is not necessary to predefine it. DEC C++ requires that the name be predefined. If you include a header file that contains any pointers to structures, you must create structure prototypes for them.

For example, consider a header file (header.h) that contains the following:

```
typedef struct  _structa {
    struct _structb    *ptr1;
    struct _structc    *ptr2;
} STRUCTA;
```

To include this header file, you must define the following structure prototypes before the #include for the header file:

```
struct  _structb;
struct  _structc;

#include <header.h> ♦
```

4.5 DEC C++ Class Library

This section contains release notes pertaining to the DEC C++ Class Library.

4.5.1 Changes and Enhancements

This section describes enhancements to the DEC C++ Class Library.

4.5.1.1 Threadsafe Support

V7.0 Starting with OpenVMS Version 7.0 and DEC C++ for OpenVMS Version 5.2, the DEC C++ Class Library is threadsafe. Refer to the *DEC C++ Class Library Reference Manual* for details about threadsafe support, including a new Mutex Package.

4.5.2 Corrections

V7.0 The following problems that existed in the DEC C++ Class Library in OpenVMS Version 6.2 have been corrected:

- When the Complex Library exp( ) function detects an underflow error, the resulting value is now (0,0) instead of (+/- max-float,) +/- max-float.
- The division routines within the Complex Library now catch divide-by-zero errors instead of signalling them.
- Starting with OpenVMS Version 7.0, the Task Package has been moved into its own shareable image, separate from the other packages within the DEC C++ Class Library. This move was made for performance reasons to remove the link between the nonthreaded library packages and DECthreads.

  This change has no effect on applications that do not use the Task Package. If your application does use the Task Package, you must relink your application on an OpenVMS Version 7.0 or later system.
- The String extraction operator now dynamically allocates the String to accommodate the input.
- When you open a file by specifying ios::ate, but not ios::app, to the filebuf open( ) function, the file is no longer opened in O_APPEND mode. Previously, this incorrect behavior caused all data to be written to the end of the file, regardless of the current file position.

4.6 DEC C Run-Time Library

The following sections contain release notes pertaining to the DEC C Run-Time Library (RTL).

4.6.1 Changes and Enhancements

This section describes changes and enhancements that are included in Version 5.2 of the DEC C RTL software. For more details, see the revision of the *DEC C Run-Time Library Reference Manual for OpenVMS Systems* that ships with DEC C Version 5.2.

4.6.1.1 Mapping of Record Carriage Control Data

V7.0 There are many improvements in the way the DEC C RTL performs I/O to RMS files with carriage-control attributes of Fortran or Print. In addition to fixing several bugs, the RTL now does a better job of mapping lines represented by the carriage-control information in the file to and from C lines. These improvements make it easier to write C programs that read or write files that are also accessed by Fortran or COBOL. However, some programs that open Fortran or Print attribute files may behave differently.

When reading a Fortran carriage-control file, the DEC C RTL now maps the space carriage-control character to a trailing \n instead of to a prefix \n and a trailing \r. This is because the space carriage-control character is used to designate a line. On output to Fortran files, the DEC C RTL now tries to emit a record whenever it encounters a \n in the output stream. This behavior is consistent with other carriage control formats. To emulate the old behavior, specify "ctx=xplct" when the file is opened, and call `fflush` when you wish to emit a record.

The new RTL behavior is fully described in the revision of the *DEC C Run-Time Library Reference Manual for OpenVMS Systems* that ships with DEC C Version 5.2.

4.6.1.2 Open File Limit Increased

V7.0

The DEC C RTL has raised the open file limit (that is, the number of files that can be open simultaneously) from 256 to 65535 on OpenVMS Alpha and to 2047 on OpenVMS VAX. The new limits are based on the OpenVMS SYSGEN parameter CHANNELCNT, which specifies the number of permanent I/O channels available to the system.

Sufficient resources must be allocated to make use of the extended file limit. Pay particular attention to the following process quotas: buffered I/O byte count, paging file, enqueue quota, and open file quota.

This support required a change to the FILE type defined in <stdio.h>. This change and documentation changes are available by upgrading to DEC C Version 5.2.

The increased open file limits impact DEC C RTL routines that deal with file descriptors or file pointers that return an error status of EMFILE (too many open files) or EBADF (bad file number) if too many files are opened. Old images running on Version 7.0 systems expect to receive a status from C RTL routines indicating "too many open files" when the old limit of 256 is reached. Now that the limits have been raised, these programs do not receive this expected status.

4.6.1.3 Support Increased for Number of Open Sockets

V7.0

The number of sockets supported by the socket routines has been increased so that it is limited only by the number of simultaneous open files allowed by the DEC C RTL (see Section 4.6.1.2).

The behavior of the socket routines (that is, those described in the DEC C Socket Routine reference section of the *DEC C Run-Time Library Reference Manual for OpenVMS Systems*) reflects support for this increased number of open files. Note that in order for the socket routines to use an increased number of sockets, the DEC TCP/IP Services for OpenVMS being used with the DEC C RTL must offer the same support.

4.6.1.4 New DEC C RTL Functions

V7.0

OpenVMS Version 7.0 introduces 122 new DEC C RTL functions that have been added to fulfill the requests of application developers and to implement the functions defined by ISO/C Amendment 1. These new functions are documented in the *DEC C Run-Time Library Reference Manual for OpenVMS Systems* distributed with DEC C for OpenVMS Systems Version 5.2.

```
basename()
bcmp()
bcopy()
btowc()
bzero()
closedir()
confstr()
dirname()
drand48()
endhostent()
endnetent()
endprotoent()
endservent()
erand48()
ffs()
fpathconf()
ftruncate()
ftw()
fwide()
fwprintf()
fwscanf()
getclock()
getdtablesize()
gethostent()
getitimer()
getlogin()
getpagesize()
getpwnam()
getpwuid()
getservent()
gettimeofday()
herror()
hostalias()
hstrerror()
index()
initstate()
ioctl()
jrand48()
lcong48()
lrand48()
mbrlen()
mbrtowc()
mbsinit()
mbsrtowcs()
memccpy()
mkstemp()
mmap()
mprotect()
mrand48()
msync()
munmap()
nrand48()
opendir()
pathconf()
pclose()
popen()
putenv()
random()
readdir()
rewinddir()
rindex()
rmdir()
seed48()
seekdir()
setenv()
sethostent()
setitimer()
setnetent()
setprotoent()
setservent()
setstate()
sigaction()
sigaddset()
sigdelset()
sigemptyset()
sigfillset()
sigismember()
siglongjmp()
sigmask()
sigpending()
sigprocmask()
sigsetjmp()
sigsuspend()
socket_fd()
srand48()
srandom()
strcasecmp()
strdup()
strncasecmp()
strsep()
swab()
swprintf()
swscanf()
sysconf()
telldir()
tempnam()
towctrans()
truncate()
tzset()
ualarm()
uname()
unlink()
unsetenv()
usleep()
vfwprintf()
vswprintf()
vwprintf()
wait3()
wait4()
waitpid()
wcrtomb()
wcsrtombs()
wcsstr()
wctob()
wctrans()
wmemchr()
wmemcmp()
wmemcpy()
wmemmove()
wmemset()
wprintf()
wscanf()
```

Applications that have implemented private versions of any of these functions are cautioned not to change over to using the DEC C RTL version prior to upgrading to DEC C Version 5.2, which contains both the header file support and documentation for these functions.

Applications that compile using the /PREFIX=ALL qualifier might get duplicate global definitions if these new symbols are present in the DECC$SHR.EXE shareable image. One option is to use the qualifier /PREFIX=(ALL,EXCEPT=*xxx*), where *xxx* represents the name of the multiply-defined function.

4.6.1.5 ICONV Utility Enhancements

V7.0 Several enhancements were made to the ICONV utility:

- The behavior of the functions `iconv`, `iconv_open`, and `iconv_close` were enhanced to allow executable-based user-defined iconv converters in addition to the current table-based method.
- The DECHANYU to EUCTW converter was made an executable converter.
- The following converters were added:
  - SDECKANJI to ISO2022JP
  - DECKANJI to ISO2022JP
  - ISO2022JP to SDECKANJI
  - ISO2022JP to DECKANJI

4.6.1.6 Specifying All Versions of File in remove( )

V7.0

The DEC C RTL remove and delete functions now accept a file specification that includes a wildcard in its version number. For example, files of the form "filename.txt;*" can be deleted by either function. Note that only the version number can be wildcarded in these functions.

4.6.1.7 Date and Time Support Changes

V7.0

Starting with OpenVMS Version 7.0, the DEC C RTL is changing the default model used to implement date and time support for programs newly compiled with DEC C Version 5.2. The new model, Universal Coordinated Time (UTC), allows the DEC C RTL to implement ANSI C/POSIX functionality that it previously could not. Using the UTC time-based model also makes the DEC C RTL compatible with the behavior of the Digital UNIX and POSIX RTL time functions.

Prior to OpenVMS Version 7.0, the DEC C RTL time-related functions dealt with local time. For compatibility, programs relinking on OpenVMS Version 7.0, or compiled with DEC C Version 5.2 on OpenVMS Version 7.0 with _DECC_V4_SOURCE and _VMS_V6_SOURCE defined, will retain the date/time behavior that existed prior to OpenVMS Version 7.0. UTC-based time functions begin with decc$__utc_*. Functions with dual time implementations are time( ), ftime( ), gmtime(), localtime( ), ctime( ), mktime( ), fstat( ), stat( ), strftime( ) and wcsftime( ).

In the UTC-based model, times are represented as seconds since the Epoch. The Epoch is defined as the time 0 hours, 0 minutes, 0 seconds, January 1, 1970 UTC. Seconds since the Epoch is a value interpreted as the number of seconds between a specified time and the Epoch.

This means that time and ftime return the time in terms of seconds since the Epoch. ctime, gmtime, and localtime take a time value representing the time in seconds from the Epoch as their argument. mktime converts a broken-down time, expressed as local time, into a time value in terms of seconds since the Epoch. The st_ctime, st_atime, and st_mtime returned in the stat structure by the stat and fstat routines are also expressed in terms of UTC. New time support includes the addition of tzset, gettimeofday, getclock, tzname, timezone, and daylight.

Due to this change, the DEC C RTL can now implement the ANSI C gmtime function, which returns a structure filled in in terms of GMT time, specify the ANSI tm_isdst field of the tm structure which specifies whether daylight savings time is in effect, and provide the POSIX and X/Open extensions to the time-related functions (tzset, tzname, timezone and daylight). In addition, the DEC C RTL can correctly compute the local time for times in the past, which is something that the time functions like localtime need to do. This could not be done using a local-time based scheme because when given a time in seconds representing a past local time, there was no way to determine the role of any local time zone information (such as whether daylight savings time was in effect).

The tzset routine gives users the ability to get time information from any time zone they wish. In addition, the localtime and gmtime routines, through the use of feature test macros, can return two additional fields, tmzone (an abbreviation of the time-zone name) and tmgmtoff (the offset from UTC in seconds), in the tm structure they return.

UTC is an international standard for measuring time of day. Under the UTC time standard, zero hours occurs when the Greenwich Meridian is at midnight. It has the advantage of always increasing, unlike local time, which can go backward or forward depending on daylight savings time. UTC also has two additional components. One, optional, is a measure of inaccuracy. The second is a time differential factor that is an offset applied to UTC to derive the local time. The time differential factor associates each local time zone with UTC. (Local times can vary up to -12 hours west of the Greenwich Meridian and +13 hours east of it).

In order for the DEC C RTL time support to work correctly in OpenVMS Version 7.0, your OpenVMS system must be correctly configured to use a valid OpenVMS TDF. Check the value of the logical SYS$TIMEZONE_DIFFERENTIAL to verify that it contains the time difference between the local time of your system and UTC. (For more details, see the section on setting up your system to compensate for different time zones in the *OpenVMS Version 7.0 New Features Manual* or in the revised *OpenVMS System Manager's Manual: Essentials* that ships on the Version 7.0 Documentation CD–ROM.)

In summary, the DEC C RTL time support is now UTC-based. Local times are computed by the DEC C RTL by applying a time-zone differential to UTC. Time-zone differential factors are supplied by default by a series of public domain source files shipped with OpenVMS or by a user locally setting a DEC C RTL environment variable by using `tzset`. For the DEC C RTL OpenVMS Version 7.0 time support to work properly, an OpenVMS Version 7.0 installation must correctly set the OpenVMS TDF and local time zone.

4.6.1.8 Message Specifications Relaxed

V7.0

The GENCAT utility has relaxed the XPG4 definition of a message line found in message catalogs. If the message text is delimited by quotation marks ("), these characters now delimit the message text. Prior to this change, the quotation marks would be included as part of the message text.

4.6.1.9 ISO Amendment 1 Changes to fscanf( ) Functions

V7.0

Starting with OpenVMS Version 7.0, the `fscanf` family of functions (`fscanf`, `scanf`, `sscanf`) change the way they handle error conditions. These changes conform to the behavior specified by Amendment 1 of the ISO C Standard or to the behavior specified by X/Open CAE Specification, Issue 4, Version 2. The error handling behavior of the `fscanf` routines has changed in the following ways:

- Starting with OpenVMS Version 7.0, an encoding error for the %S and %C directives is considered an input failure. Previously, an encoding error was considered a matching failure in these cases.
- Starting with OpenVMS Version 7.0, if at least one input character is successfully read for a %S or %C directive, the directive is not considered to fail. Previously, an encoding error encountered while processing a %S or %C directive caused the entire directive to fail even if several (multibyte) characters were successfully read and copied into the target array. For example, prior to OpenVMS Version 7.0, the following invocation of `sscanf` would return "0" if the multibyte character \x8e\x20 is illegal in the current program's locale:

  ```
  sscanf ("a\x8e\x20", "%2S", ws)
  ```

  Starting with OpenVMS Version 7.0, this invocation returns "1".

- Starting with OpenVMS Version 7.0, the `fscanf` family of functions sets the error indicator on a stream when a conversion error occurs. Previously, if an encoding error occurred, an error indicator for the stream was not set. (This conforms to X/Open CAE Specification, Issue 4, Version 2).

4.6.1.10 New 4.4 BSD Curses Functions (Alpha Only)

V7.0

Alpha

OpenVMS Alpha 7.0 reintegrates the BSD curses package with sources distributed with 4.4 BSD. The following 31 additional functions were added to the list of BSD curses functions. These functions are defined in the `<curses.h>` header file distributed with DEC C for OpenVMS Systems Version 5.2. The function names of the form _xxx64 are 64-bit pointer support for those functions.

```
baudrate()
cbreak()
echo()
erasechar()
fullname()
idlok()
killchar()
nl()
nocbreak()
nodelay()
noecho()
nonl()
noraw()
raw()
resetty()
savetty()
tgetent()
tgetflag()
tgetnum()
tgetstr()
tgoto()
touchline()
touchoverlap()
tputs()
waddnstr()
wattroff()
wattron()
wattrset()
```
♦

4.6.1.11 Internationalization Support

V7.0 The DEC C RTL has added capabilities to allow application developers to create international software. The DEC C RTL obtains information about a language and a culture by reading this information from *locale* files.

If you are using these DEC C RTL capabilities, you must install a separate kit to provide these files to your system. The save set, VMSI18B070, is provided on the same media as OpenVMS Version 7.0.

To install this save set, follow the standard OpenVMS installation procedures using this save set name as the name of the kit. There are four categories of locales that you can select to install. You may select as many locales as you need by answering the following prompts:

```
Do you want European and US support?
Do you want Chinese support?
Do you want Japanese support?
Do you want Korean support?
```

This kit also has an Installation Verification Procedure that Digital recommends you run to verify the correct installation of the kit.

4.6.1.12 LOCALE Utility Change

V7.0 To conform with ISO/IEC 9899-1990:1994 (the amendment to the ISO C Language standard), the locale definition utility no longer permits any character from the space class to be in the graph or punct classes.

4.6.1.13 Floating-Point Values Now Conform to IEEE Standard

V6.2 The `ecvt`, `fcvt`, and `gcvt` functions were modified to represent the following values specified in the IEEE Standard for floating-point arithmetic:

Value	Representation
Quiet NaN	NaNQ
Signaling NaN	NaNS
+Infinity	Infinity
-Infinity	-Infinity

The sign associated with each of these values is stored in the sign parameter of ecvt, fcvt, and gcvt. Also note that in IEEE Floating Point, a value of zero can be positive or negative, as set by the sign parameter.

4.6.1.14 Math Functions Conform to XPG4 (VAX Only)

V6.2

VAX

The following changes have been made to DEC C RTL in OpenVMS VAX Version 6.2 so that the behavior now matches that on OpenVMS Alpha systems.

- The behavior of the log, log10, and pow functions now conforms to the behavior specified by XPG4. As such, log and log10 of zero return -HUGE_VAL and set errno to ERANGE; pow of zero with a negative exponent returns -HUGE_VAL and sets errno to EDOM; and pow of zero with a zero exponent returns 1.0. ♦

4.6.1.15 Open Routines Now Validate RMS Options

V6.2 The open, fopen, and creat functions no longer ignore invalid RMS options. If an unrecognized RMS option is passed as a parameter to these functions, a return value of -1 is returned and errno is set to EINVAL (Invalid argument).

4.6.1.16 Setvbuf Function

V6.2 The setvbuf function has been changed to give it more functionality and make it more compatible with the setvbuf function on other operating systems.

Prior to this release, if the buffer argument to setvbuf was NULL, setvbuf would reset the buffer to the one allocated by the DEC C RTL when the file was opened. Starting with Version 6.2, if a NULL buffer argument is passed and the size argument is larger than the buffer allocated by the RTL when the file was opened, setvbuf allocates a buffer equal to the specified size and uses that as the file buffer.

4.6.1.17 Sleep Function Now Returns Unslept Seconds

V6.2 The return value of the sleep function has been changed to be in line with the System V Interface Definition and the X/Open Portability Guide, Release 4. The return value is the number of seconds that it prematurely awoke, due to the delivery of a signal or a call from the sys$wake function. A return value of zero is returned when a sleep call waits the entire time period. Prior to these changes, the sleep function returned the number of seconds that were originally requested.

4.6.1.18 Wildcards No Longer Accepted by stat Function

V6.2 The stat function no longer accepts wildcard characters as part of the file specification. If a wildcard character is found, stat returns -1 and sets errno to ENOENT (No such file or directory).

4.6.1.19 Writing to Pipes Now Conforms to POSIX Standards

V6.2 The DEC C RTL now properly deals with EOF when reading and writing to pipes. The correct behavior is described in the IEEE standard P1003.1-1988, Section 6.4.1.2, which states that an EOF is written to a pipe only when no writers are left. Previously, the DEC C RTL wrote an EOF to a pipe every time a writer closed a pipe.

The DEC C RTL no longer incorrectly writes an EOF to a pipe when the user issues a write of zero bytes. The corrected behavior now conforms to the IEEE Standard P1003.1-1988, Section 6.4.1.2, which states that writing zero bytes to a file has no effect.

4.6.2 Problems and Restrictions

This section describes problems and restrictions with DEC C RTL software.

4.6.2.1 Socket Behavior Prior to UCX V3.3

V7.0 In programs that link with a version of the DEC C RTL that supports passing sockets from a parent process to a child process and that links with UCX Versions 3.2 or less, a parent process cannot shut down or close a socket that has been closed by a child process, either specifically by the child or by default when the child terminates. With these versions of UCX, closing a socket breaks the TCP connection and another process cannot send any more data to it. For the `shutdown` function, the error status returned is EINVAL "invalid argument". For the `close` function, errno indicates EVMSERR.

Additionaly, when a parent closes a socket, the socket is no longer available to any child processes that have inherited the socket in programs linked as specified above. On UCX Version 3.2 and lower, a parent's closure of a socket interrupts operations pending from a child process on the same socket. In this case, a parent process should await termination of all its children before closing its sockets, and then ignore any errors when it does close them.

4.6.2.2 Linking DEC C Applications Using /NOSYSSHR

V7.0 When linking DEC C programs against the DEC C RTL object libraries using the /NOSYSSHR qualifier, applications that previously linked without undefined globals may now result in undefined globals for the CMA$TIS symbols. To resolve these undefined globals, add the following line to your link options file:

```
SYS$LIBRARY:STARLET.OLB/LIBRARY/INCLUDE=CMA$TIS
```

4.6.2.3 Linking With /NOSYSSHR Disables Dynamic Activation

V7.0 If a program linked with the /NOSYSSHR qualifier makes a call to a routine that resides in a dynamically activated image, the routine returns a value indicating an unsuccessful status, errno is set to ENOSYS, and vaxc$errno is set to C$_NOSYSSHR. The error message corresponding to C$_NOSYSSHR is "Linking /NOSYSSHR disables dynamic image activation". An example of this situation is a program linked with /NOSYSSHR that makes a call to a socket routine.

4.6.2.4 Internationalization Compatibility Problem with DECwindows Motif

V6.2 Applications that call the Xlib locale routines in DECwindows Motif Version 1.2–3 using the method described in Section 4.18.5.3 of the *DECwindows Motif Version 1.2–3 for OpenVMS Release Notes* will continue to function on OpenVMS Version 6.2. However, because the locale support in Xlib is not compatible with the support in the DEC C Run-Time Library (DECC$SHR.EXE), Xlib does not use the locale environment provided by the C library. Therefore, setting the locale in the C library does not affect the behavior of DECwindows Motif, although it does affect C library routines such as strcoll( ). Setting the locale in Xlib changes the behavior of DECwindows Motif but does not affect C library routines.

Applications that call the Xlib locale routines in DECwindows Motif Version 1.2 using the method described in the *DECwindows Motif Version 1.2–3 for OpenVMS Release Notes*, Section 4.17.6.3, do *not* run correctly on OpenVMS Version 6.2 unless you install the DECwindows Motif Version 1.2 remedial kit appropriate for your system. Contact your Digital support representative for the following kits that add support for the DECW$USEXLIBXPG4 logical name:

	Kit Number	
Platform	**For Base Product**	**For WorldWide Support**
Alpha	AXPMOTF05_012	DWMA_I18N01_012
VAX	VAXMOTF05_012	DWMV_I18N01_012

With the remedial kit installed, the Xlib locale routines work normally, but Xlib does not use the locale environment provided by the C library.

Digital expects to correct this problem in a future release of DECwindows Motif.

4.6.3 Corrections

This section describes corrections to the DEC C RTL software.

4.6.3.1 Open Sockets Passed to Child Processes

V7.0 Socket connection information is now passed from a parent to a child process when using `vfork/exec` functions that pass open file information to the child process.

4.6.3.2 File Protection Status Now Mapped to EACCES

V7.0 For compatibility with most UNIX implementations, the error status RMS$_PRV (insufficient privilege or file protection violation) is now mapped to an errno value of EACCES (permission denied). Previously, it was mapped to EPERM (not owner).

4.6.3.3 Invalid RMS Options Cause Open to Fail

V7.0 DEC C RTL functions that can be passed RMS options now validate that these options are valid RMS options. These functions no longer ignore invalid options, but instead return failure.

4.6.3.4 LOCALEDEF Utility

V7.0 The LOCALEDEF utility now consistently verifies that the locale entities *decimal_point*, *thousands_sep*, and *mon_thousands_sep* are (multibyte) strings, containing one (multibyte) character.

4.6.3.5 UNIX File Specifications Containing Relative Paths

V7.0 OpenVMS file specifications representing relative pathnames are now translated into UNIX style pathnames in the following way: an OpenVMS file specification "[.x.y]" representing a relative pathname is translated into "./x/y". Previously, such an OpenVMS file specification was translated into "/x/y".

4.6.3.6 Behavior of realloc( ) Function

V7.0

Calls to the realloc function of the form realloc(*address*,0) now free the memory pointed to by the *address* parameter as specified by the ANSI C standard.

4.6.3.7 Usage of SYS_ERRLIST and SYS_NERR

V7.0

In OpenVMS Version 7.0, the DEC C RTL exports the global symbols sys_nerr (the maximum errno value) and sys_errlist (the array of message strings that corresponds to the list of errors specified in <errno.h>). These symbols are provided for compatibility purposes only. The recommended way to determine the last error (that is, the errno value) encountered during a call to the DEC C RTL from a C program, is by use of the perror and strerror functions.

These symbols were first exported globally by the DEC C RTL in OpenVMS Version 6.2. Several regressions in the use of sys_nerr and sys_errlist caused by this release are corrected in OpenVMS Version 7.0. The regressions are as follows:

1. On OpenVMS Alpha Version 6.2, unprefixed symbols for sys_nerr and sys_errlist were removed from SYS$LIBRARY:STARLET.OLB. These symbols were removed because DEC C symbols in STARLET.OLB should be prefixed by DECC. However, removing these symbols resulted in undefined symbols when previously compiled programs referencing sys_nerr and sys_errlist relinked on OpenVMS Alpha Version 6.2. The resulting linker warnings were similar to the following:

```
%LINK-W-NUDFSYMS, 2 undefined symbols:
%LINK-I-UDFSYM,         SYS_ERRLIST
%LINK-I-UDFSYM,         SYS_NERR
%LINK-W-USEUNDEF, undefined symbol SYS_NERR referenced
        in psect $LINK$ offset %X000000A0
        in SNLER FOO file device:[directory]filename.type;version
%LINK-W-USEUNDEF, undefined symbol SYS_ERRLIST referenced
        in psect $LINK$ offset %X000000A8
        in module SNLER file device:[directory]filename.type;version
```

2. Using the DEC C Version 4.1 (OpenVMS Alpha only) or DEC C Version 5.0 compiler (OpenVMS Alpha/VAX) on OpenVMS Version 6.2 could cause the compilation or run-time errors described in the following code. The <perror.h> provided by DEC C Version 4.1 and DEC C Version 5.0 defined sys_nerr and sys_errlist on OpenVMS Version 6.2 and higher in the following way:

```
#   pragma __extern_model __save
#   pragma __extern_model __relaxed_refdef
#   define sys_nerr decc$gl_sys_nerr
#   define sys_errlist decc$ga_sys_errlist
    extern int decc$gl_sys_nerr;
    extern char **decc$ga_sys_errlist;
#   pragma __extern_model __restore
```

The problems with this definition are:

- Programs that define sys_errlist internally as follows do not recompile if they also include the DEC C Version 4.1 or DEC C Version 5.0 <perror.h>:

```
char * sys_errlist []
```

- Programs that define sys_errlist internally as follows, but do not include <perror.h>, will recompile. However, they will cause an access violation at run time.

  ```
  char * sys_errlist []
  ```

These regressions were fixed in OpenVMS Version 7.0 by doing the following:

- Replacing the unprefixed sys_nerr and sys_errlist symbols back into STARLET.OLB.
- Changing the definition of sys_errlist back to char *sys_errlist[] rather than defining it as char **sys_errlist.

The following sections describe how to use sys_nerr and sys_errlist correctly on OpenVMS Version 7.0.

4.6.3.7.1 Recompiling with DEC C Version 4.1 or Version 5.0 (Alpha Only)

V7.0

Alpha

If you have installed OpenVMS Alpha DEC C Version 4.1 or Version 5.0, new compilations on OpenVMS Version 7.0 of programs using sys_nerr and sys_errlist require the version of <perror.h> that shipped with DEC C Version 5.2. You must either create a new <perror.h> or edit an existing file so that it contains the definitions of sys_nerr and sys_errlist supplied with DEC C Version 5.2. Use the following steps to reconstruct a new <perror.h> every time a DEC C Version 4.1 or Version 5.0 compiler is reinstalled on an OpenVMS Version 7.0 system:

1. Create a version of <perror.h> containing the following contents:

   ```
   #ifndef __PERROR_LOADED
   #define __PERROR_LOADED 1
   /****************************************************************************
   **
   **  <perror.h> - error message variables
   **
   **  Copyright Digital Equipment Corporation 1993, 1995. All rights reserved.
   **
   **  Restricted Rights: Use, duplication, or disclosure by the U.S.
   **  Government is subject to restrictions as set forth in subparagraph
   **  (c) (1) (ii) of DFARS 252.227-7013, or in FAR 52.227-19, or in FAR
   **  52.227-14 Alt. III, as applicable.
   **
   **  This software is proprietary to and embodies the confidential
   **  technology of Digital Equipment Corporation. Possession, use, or
   **  copying of this software and media is authorized only pursuant to a
   **  valid written license from Digital or an authorized sublicensor.
   **
   *****************************************************************************
   */
   #pragma __nostandard
   #ifdef __cplusplus
      extern "C" {
   #endif

   /*
   **  sys_nerr and sys_errlist are implemented as of OpenVMS V6.2
   */
   #if __VMS_VER >= 60200000
   ```

```
   /*
   **  Real values in the DECC$SHR image
   */
#   pragma ___extern_model ___save
#   pragma ___extern_model ___relaxed_refdef
    extern int sys_nerr;
    extern char *sys_errlist[];
#   pragma ___extern_model ___restore

#else

   /*
   **  Pre-V6.2 Stub values resolved in OLBs
   */
#ifdef ___ALPHA
#   pragma ___extern_model ___save
#   pragma ___extern_model ___relaxed_refdef
    extern int sys_nerr;
    extern char *sys_errlist[];
#   pragma ___extern_model ___restore
#else
#   pragma ___extern_model ___save
#   pragma ___extern_model ___common_block
    extern int sys_nerr = 0;
    extern char *sys_errlist[];
#   pragma ___extern_model ___restore
#endif
#endif

#ifdef ___cplusplus
    }
#endif
#pragma ___standard
#endif /* ___PERROR_LOADED */
```

2. Save the version of <perror.h> provided with your version of the DEC C compiler as follows:

   ```
   $ LIBRARY/EXTRACT=PERROR/OUT=SAVED_PERROR.H SYS$LIBRARY:DECC$RTLDEF.TLB
   ```

3. Replace the new <perror.h> for future use as follows. Note that DIRECTORY_SPEC represents the directory specification for the location of the new version of <perror.h>.

   ```
   $ LIBRARY/REPLACE SYS$LIBRARY:DECC$RTLDEF.TLB DIRECTORY_SPEC:PERROR.H ♦
   ```

4.6.3.7.2 Recompiling Pre-Version 4.1 (OpenVMS Alpha) or Pre-Version 5.0 (OpenVMS VAX) DEC C Compilers

V7.0 If you have a pre-Version 4.1 (OpenVMS Alpha) or pre-Version 5.0 (OpenVMS VAX) DEC C compiler installed on an OpenVMS Version 7.0 system and plan to compile programs using sys_nerr and sys_errlist, you must determine how these symbols are defined by the <perror.h> provided with your DEC C compiler. To do this, extract the current version of <perror.h> as:

```
$ LIBRARY/EXTRACT=PERROR/OUT=PERROR.H SYS$LIBRARY:DECC$RTLDEF.TLB
```

Examine <perror.h> to determine if sys_nerr and sys_errlist are defined using the common block extern model. If this is the case, you need to link a program that uses these symbols in the following way:

```
$ LINK PROGRAM,SYS$LIBRARY:STARLET/INC=(C$SYS_ERRLIST_UP,C$SYS_ERRLIST_LP)
```

If these symbols are defined using the relaxed refdef extern model, no special linking instructions are required. The reason for this is that pre-DEC C Version 4.1 and pre-DEC C Version 5.0 compilers on OpenVMS 7.0 cause references to sys_nerr and sys_errlist to be prefixed with DECC. References to prefixed sys_nerr and sys_errlist names that are defined using the relaxed refdef extern model are resolved in the OpenVMS Version 7.0 DEC C RTL shareable image or in STARLET.OLB. References to prefixed sys_nerr and sys_errlist names that are defined using the common block extern model are resolved as prefixed psects in the modules C$SYS_ERRLIST_UP and C$SYS_ERRLIST_LP residing in STARLET.OLB. C$SYS_ERRLIST_UP contains uppercase versions of sys_nerr and sys_errlist, and C$SYS_ERRLIST_LP contains lowercase versions of sys_nerr and sys_errlist.

4.6.3.7.3 Relinking Programs Compiled on Earlier Versions of OpenVMS

V7.0 The following guidelines should enable the recompilation of programs using sys_nerr and sys_errlist on OpenVMS Version 7.0 and the relinking of programs compiled on previous versions of OpenVMS.

- Programs compiled on OpenVMS Version 6.2 using DEC C Version 5.0 or DEC C Version 4.1 can be relinked on OpenVMS Version 7.0 without special instructions.
- Previously compiled programs that make references to unprefixed symbols for sys_nerr and sys_errlist should relink on OpenVMS Version 7.0 without special instructions. Unprefixed symbols for sys_nerr and sys_errlist are again resolved in STARLET.OLB on OpenVMS Version 7.0.
- Previously compiled programs that make references to unprefixed psects for sys_nerr and sys_errlist can be relinked as described below. Note that these types of programs are programs compiled on OpenVMS VAX with a pre-DEC C Version 5.0 compiler and on OpenVMS Alpha with a Version 1.3 DEC C or DEC C++ compiler or a DEC C Version 4.0 compiler.

  ```
  $ LINK PROGRAM,SYS$LIBRARY:STARLET/INC=(C$SYS_ERRLIST_U,C$SYS_ERRLIST_L)
  ```

  The modules C$SYS_ERRLIST_U and C$SYS_ERRLIST_L contain unprefixed definitions for sys_nerr and sys_errlist as psects. C$SYS_ERRLIST_U contains upper case versions of sys_nerr and sys_errlist. C$SYS_ERRLIST_L contains lowercase versions of sys_nerr and sys_errlist.

4.7 DECthreads

The following sections contain release notes pertaining to DECthreads.

4.7.1 Changes and Enhancements

This section summarizes some of the significant changes and enhancements made to DECthreads in Version 7.0. For detailed information about using DECthreads, refer to the *Guide to DECthreads*.

4.7.1.1 POSIX 1003.1c Standard Style Interface

V7.0 With OpenVMS Version 7.0, the DECthreads library (PTHREAD$RTL.EXE) implements the POSIX 1003.1c standard interface (IEEE Std 1003.1c-1995, POSIX System Application Program Interface), as approved by the IEEE standards board in June 1995. The new POSIX (pthread) interface supported with DECthreads is the most portable, efficient, and powerful OpenVMS programming interface for a multithreaded environment. This interface is defined by the C language header file <pthread.h>. No exception interface exists for the POSIX 1003.1c standard style interface.

4.7.1.2 Thread Independent Services (TIS) Interface

V7.0 OpenVMS Version 7.0 includes the Thread Independent Services (TIS) application programming interface (CMA$TIS_SHR.EXE). TIS provides services that assist with the development of thread-safe application programming interfaces (APIs).

Thread synchronization can involve significant run-time cost, which is undesirable in the absence of threads. TIS enables you to build thread-safe APIs that are efficient in the nonthreaded environment, yet provide the necessary synchronization in the threaded environment.

When DECthreads is not active within the process, TIS executes only the minimum steps necessary. Code running in a nonthreaded environment is not burdened by the run-time synchronization that is necessary when the same code is run in a threaded environment. When DECthreads is active, the TIS functions provide the necessary thread-safe synchronization.

4.7.1.3 POSIX 1003.4a Draft 4 Interface Retirement

V7.0 The POSIX 1003.4a, Draft 4 interface of DECthreads will be retired in a future release. Applications that were written using the POSIX 1003.4a, Draft 4 API should be migrated to the new IEEE Std 1003.1c-1995, POSIX System Application Program Interface provided by DECthreads. A compatibility mode for the Draft 4 POSIX 1003.4a application programming interface (API) has been provided in this release to help ease migration. This compatibility mode will be removed in a future release. See the *Guide to DECthreads* for information regarding use of the source compatibility mode.

4.7.1.4 CMA Interface: Change in Status

V7.0 In future releases, the CMA interface of DECthreads will continue to exist and be supported in the OpenVMS operating system, but it will no longer be documented or enhanced. Therefore, Digital recommends that you port your CMA-based application to the IEEE Std 1003.1c-1995, POSIX System Application Program Interface provided by DECthreads.

4.7.1.5 Application Coding Errors

V7.0 OpenVMS Version 7.0 contains substantial changes to threads that will likely expose programming errors in existing applications that use DECthreads. Such errors include, but are not limited to, the following:

- Attempting to unlock a mutex that is not locked
- Use of uninitialized variables
- Improper use of data structures (for example, using a `pthread_attr_t` instead of a `pthread_mutexattr_t` in a call to `pthread_mutexattr_create`)
- Improper data access synchronization
- Use of an undocumented or unsupported routine

4.7.1.6 Timed Condition Wait Time Models

V7.0 The POSIX Draft 4 and CMA versions of the timed condition wait routines (`pthread_cond_timedwait` and `cma_cond_timed_wait`) require local time for the time argument (*abstime* and *expiration* respectively). The `pthread_cond_timedwait` provided by the new POSIX 1003.1c standard style interface requires *abstime* to be specified in Universal Coordinated Time (UTC). See documentation of `pthread_cond_timedwait` in the *Guide to DECthreads*.

4.7.1.7 $EXIT and Exit Handling Changes (Alpha Only)

V7.0

Alpha

$EXIT and exit handling for multithreaded processes has changed significantly in OpenVMS Alpha Version 7.0. Exit handlers are executed in a separate thread, not in the thread that calls the $EXIT system service routine. A call to $EXIT in a multithreaded process results in a call to `pthread_exit`. See the *Guide to DECthreads* for more information. ♦

4.7.1.8 Condition Variable Waits (Alpha Only)

V7.0

Alpha

Threaded applications on OpenVMS Alpha Version 7.0 that use the condition wait and timed condition wait routines may experience more spurious wakeups than previously. Correctly coded applications should not be affected by this change. See the *Guide to DECthreads* for more information. ♦

4.7.1.9 Behavior on Multiprocessors (Alpha Only)

V7.0

Alpha

On OpenVMS Alpha Version 7.0, DECthreads has the ability to create virtual processors as they are needed by the application. The number of virtual processors that DECthreads creates is limited by the SYSGEN parameter MULTITHREAD. This parameter is typically set to the number of processors that reside in the system. Regardless of the value of this parameter, DECthreads will create no more virtual processors than there are user threads (excluding DECthreads internal threads).

The "vp-count" DECthreads configuration option can be used to control the maximum number of virtual processors that DECthreads creates. For example, to explicitly set the number of virtual processors to 2, use the following command:

```
$ DEFINE PTHREAD_CONFIG "vp-count=2"
```

When "vp-count" is zero, the number of virtual processors that can be created is equal to whichever of the following is the lower value:

- The value of the SYSGEN parameter MULTITHREAD
- The number of physical processors in the system configuration

Note

When "vp-count" has a value other than 1, simultaneous execution of threads is possible on multiprocessor systems. In threaded applications, this can expose improper synchronization coding techniques. If this occurs, you may want to restrict your application's process to use a single thread.

DECthreads does not delete virtual processors or let them terminate. They are retained in an idle (HIB) state until they are needed again. During image rundown, they are deleted by the OpenVMS executive.

The DECthreads scheduler can schedule any user thread onto any virtual processor. Therefore, a user thread can run on different kernel threads at different times. Normally, this should pose no problem. However, you may notice small changes; for example, a user thread's PID (retrieved by querying the system) might change from time to time. ♦

4.7.1.10 Blocking System Service Calls (Alpha Only)

V7.0

Alpha With OpenVMS Alpha Version 7.0, system service calls that block are now thread synchronous; that is, they block only the calling thread instead of the entire process. See the *Guide to DECthreads* for more information. ♦

4.7.1.11 $HIBER (Alpha Only)

V7.0

Alpha OpenVMS Alpha Version 7.0 threaded applications that use $HIBER may experience more spurious wakeups than in previous releases. After the call to $HIBER returns, you must validate the wakeup to ensure that it is appropriate for execution to proceed. This wakeup validation has always been required, but it is now more important. See the *Guide to DECthreads* for more information.

Prior to Version 7.0, a thread that called $HIBER (or that called a library routine that resulted in a call to $HIBER) would cause the whole process to hibernate briefly whenever that thread was scheduled to run. With multiple threads in simultaneous calls to $HIBER, there was no reliable way to wake one or all of the threads. The next hibernating thread to be scheduled would awaken, and any other threads would continue to sleep.

In Version 7.0, these problems are resolved. However, the new behavior has other effects. For instance, hibernation-based services such as LIB$WAIT and the C RTL `sleep()` routine might complete prematurely if the service does not validate its wakeup (to ensure that enough time has passed or that there is some other reason for it to return). ♦

4.7.1.12 DECthreads Debugger

V7.0 A number of changes have been made in the DECthreads debugging features, including how to use them. These changes are unrelated to any changes in the OpenVMS Debugger. See the *Guide to DECthreads* for more information.

4.7.2 Problems and Restrictions

This section describes known DECthreads problems and restrictions.

4.7.2.1 Release Compatibility

V7.0 Binary compatibility and a source compatibility mode for the Draft 4 POSIX interface and the CMA interface to DECthreads are provided in Version 7.0. However, no object compatibility for object modules created on earlier OpenVMS releases is provided for object modules that use the Draft 4 POSIX interface.

4.7.2.2 Language Support

V7.0 This release does not include interface definitions for non-C languages for the POSIX 1003.1c standard style interface to DECthreads. All DECthreads routines are usable from non-C languages; however, the application must provide the routine declarations. These self-defined declarations should be modeled after the C language declarations in pthread.h.

Also, when using DECthreads, beware of using common language practices that are inherently not thread-safe. For example, Fortran compilers typically rely heavily on static storage. These factors need to be considered and addressed when writing threaded applications and thread-safe libraries.

Improved language support is expected to be provided in a future release.

4.7.2.3 64-Bit Exception Handling (Alpha Only)

V7.0

Alpha

No 64-bit exception handling has been provided in DECthreads for Version 7.0. This support will be provided on OpenVMS Alpha in a future release. ♦

4.7.2.4 64-Bit Parameter Support in Routines (Alpha Only)

V7.0

Alpha

The POSIX 1003.1c style interface to DECthreads supports 64-bit parameters (see the description of the `pthread_join` routine in the *Guide to DECthreads*). No 64-bit parameter support has been provided in the CMA or POSIX Draft 4 DECthreads' interfaces. These interfaces will *not* be enhanced to include 64-bit parameter support. ♦

4.7.2.5 Using Multiple Kernel Threads on Debugger (Alpha Only)

V7.0

Alpha

Using the OpenVMS Debugger with multiple kernel threads produces a number of problems on OpenVMS Alpha. When debugging, you may want to restrict your application to use one virtual processor. See Section 4.7.1.9 for virtual processor restriction information. ♦

4.7.2.6 Interoperability with POSIX for OpenVMS

V7.0

Previous releases of the POSIX for OpenVMS layered product had very limited interoperability with DECthreads. In OpenVMS Version 7.0, using DECthreads with the POSIX for OpenVMS layered product is not supported.

4.7.2.7 DECthreads Debugger Metering Function

V7.0

The metering capability of the DECthreads debugger does not work in this release.

4.7.2.8 Condition Variable Attributes Routines

V7.0

The `pthread_condattr_init` and `pthread_condattr_destroy` routines are provided for future expansion of the pthread interface and to provide as much POSIX functionality as possible for applications migrating to OpenVMS from POSIX-compliant operating systems. These routines currently serve no useful function in the OpenVMS Version 7.0 operating system because there are not yet any `pthread_condattr_set*` type routines.

4.7.2.9 C Run-Time Library *errno* Value

V7.0

When *errno* is accessed from the OpenVMS Debugger, the value of the global errno (not the per-thread errno) is returned. (This is not a new condition; it just has not been documented previously.)

4.7.2.10 SET TASK/ACTIVE Command

V6.2

The OpenVMS Debugger command SET TASK/ACTIVE does not work for DECthreads (on both OpenVMS Alpha and VAX systems) or for DEC Ada for OpenVMS Alpha systems, the tasking for which is implemented using DECthreads.

Instead, you can use the following effective substitutes on DECthreads:

- For query-type actions, use the SET TASK/VISIBLE command.
- To gain control to step through a particular thread, use strategic placement of breakpoints.

4.7.2.11 Dynamic Image Activation

V6.2

Applications that use thread-safe run-time libraries might not be able to use LIB$FIND_IMAGE_SYMBOL to dynamically activate DECthreads or products that depend on DECthreads.

Certain run-time libraries use conditional synchronization mechanisms. These mechanisms typically are enabled during image initialization when the run-time library is loaded only if the process is multithreaded. If the process is not multithreaded, the synchronization is disabled.

If a single-threaded application attempts to use LIB$FIND_IMAGE_SYMBOL to activate an image that would cause the process to become multithreaded, a LIB$_REENTRANCY error occurs and the application is terminated.

This termination ensures that threads can never be present when run-time libraries are not using the proper synchronization.

To work around this problem, link the image that calls LIB$FIND_IMAGE_SYMBOL against PTHREAD$RTL.

4.7.2.12 Exit Handler Routine (VAX Only)

V7.0

Note

Prior to Version 7.0, this note applied to both VAX and Alpha. The behavior described in this note now applies to VAX only. See Section 4.7.1.7 for a description of changes to $EXIT and exit handling in OpenVMS Alpha Version 7.0.

V6.1

VAX

If you try to abort a program that uses DECthreads functions in an exit handler routine by using a Ctrl/Y sequence followed by the DCL command EXIT (or almost any DCL command), the program may behave incorrectly; for example, it might hang indefinitely in the exit handler routine.

One instance of this problem occurs when you type a Ctrl/Y sequence to interrupt a multithreaded program in the middle of a C RTL I/O function. The problem is with the operating system, not with your program code or the C Run-time Library.

To release your program from the exit handler routine, type another Ctrl/Y sequence followed by the DCL command STOP.

Digital expects to correct this problem in a future release. ♦

4.7.2.13 DECthreads Exit Routines Can Terminate Process (VAX Only)

V6.0

VAX

The DECthreads routines `cma_thread_exit_error`, `cma_thread_exit_normal`, and `pthread_exit` should terminate the calling thread only. However, on OpenVMS VAX, these routines erroneously cause the process to terminate when they are called in the initial thread.

This problem will be corrected in a future release. ♦

4.7.3 Documentation Changes and Corrections

The following note describes a change in the DECthreads online help.

4.7.3.1 Online Help

V7.0

No online help has been provided for DECthreads in OpenVMS Version 7.0. It is expected that DECthreads help will be provided in a future release.

4.8 DECTPU for DECwindows Motif

The following sections contain release notes pertaining to DECTPU for DECwindows Motif.

4.8.1 Problems and Restrictions

This section describes DECTPU for DECwindows Motif problems and restrictions.

4.8.1.1 Small Display Monitors and DECwindows Motif Applications

V6.0

When running DECwindows Motif DECTPU on small display monitors, the main window can be less than fully visible.

To correct this condition, follow these steps:

1. Add the following resources to your DECTPU X resource file:

```
Tpu.Tpu$MainWindow.X:                              0
Tpu.Tpu$MainWindow.Y:                              0
Tpu.Tpu$Mainwindow.Rows:                           21
Tpu*condensedFont:                                 on
Tpu*fontSetSelection:                              1
```

2. Copy the resource file from `SYS$LIBRARY:EVE.DAT` and add the previous lines.
3. Use logical name TPU$DEFAULTS to point at the new resource file.

   The following example invokes the EVE DECwindows Motif user interface using the X resource file named eve_small_window.dat in your login directory to edit the file LOGIN.COM.

```
$ DEFINE TPU$DEFAULTS SYS$LOGIN:EVE_SMALL_WINDOW.DAT
$ EDIT/TPU/INTER=DECWINDOWS LOGIN.COM
```

4.9 Delta/XDelta Debugger

The following sections contain release notes pertaining to the Delta/XDelta debugger.

4.9.1 Changes and Enhancements

This section describes changes or enhancements to the software.

4.9.1.1 Image Activation (Alpha Only)

V6.1

Alpha

Because of enhancements to the OpenVMS Debugger, the image activator on OpenVMS Alpha systems has been modified to automatically activate SYS$SHARE:SYS$SSISHR.EXE when an image is debugged using the RUN /DEBUG command or is linked using the /DEBUG qualifier.

If the Delta/XDelta debugger is being used, SYS$SHARE:SYS$SSISHR.EXE may be automatically activated for you. The presence of this image should not alter your program's correctness, but if your program is sensitive to virtual address layout or if for some reason SYS$SHARE:SYS$SSISHR.EXE is not installed properly on your system, you may want to bypass its automatic activation.

To keep the image activator from activating SYS$SHARE:SYS$SSISHR.EXE for you, define the logical name SSI$AUTO_ACTIVATE to be "OFF" before running the program to be debugged with Delta/XDelta. ♦

4.9.2 Documentation Changes and Corrections

This section describes corrections to the Delta/XDelta documentation.

4.9.2.1 *OpenVMS Delta/XDelta Debugger Manual*

The following notes describe corrections to the *OpenVMS Delta/XDelta Debugger Manual*.

4.9.2.1.1 Floating Point Register Support (Alpha Only)

V7.0

Alpha

In the *OpenVMS Delta/XDelta Debugger Manual*, the statement immediately following Table 2-1, "On AXP, note that the floating point registers cannot be accessed by DELTA or XDELTA," is incorrect. On OpenVMS Alpha, floating point registers can be accessed from DELTA and from XDELTA but only if floating point arithmetic is enabled in the current process.

DELTA runs in the context of a process. Access to floating point registers is enabled as soon as the first floating point instruction in the code being examined is executed. Access is disabled as soon as that image completes execution.

When the system enters XDELTA, it may not be obvious which process is the current process. If the current process happens to have floating point enabled (because a floating point instruction has executed and the image containing the floating point instruction is still executing), then you can access the floating point registers. Otherwise, you cannot. XDELTA checks the FEN (floating point enable) IPR (internal processor register) to see whether it needs to provide access to floating point registers. ♦

4.9.2.1.2 Base Register Default Offset (Alpha Only)

V1.5

Alpha

The base register default offset for OpenVMS Alpha is 10000_{16}, not 100000_{16}, as documented in the description of the ;X (Load Base Register) command. ♦

4.10 Device Support on OpenVMS Alpha Systems

Alpha

The following sections contain release notes pertaining to OpenVMS Alpha device drivers.

4.10.1 Changes and Enhancements

This section describes OpenVMS Alpha device support changes or enhancements.

4.10.1.1 OpenVMS Alpha Device Drivers Must Be Recompiled and Relinked

V7.0

OpenVMS Alpha Version 7.0 includes significant changes to OpenVMS Alpha privileged interfaces and data structures. As a result of these changes, all device drivers from previous versions of OpenVMS Alpha must be recompiled and relinked to run correctly on OpenVMS Alpha Version 7.0.

For more details about OpenVMS Alpha Version 7.0 changes that may require source changes to customer-written drivers, see the *OpenVMS Alpha Guide to Upgrading Privileged-Code Applications*.

4.10.1.2 Bus Array Entry Structure Change

V7.0

The bus array entry structure has been expanded for OpenVMS Alpha Version 7.0. The new bus array entry is one quadword larger than the old one.

This change requires any code which walks the bus array (for example, ICBMs used to auto-configure device drivers) to recompile and relink to run on OpenVMS Alpha Version 7.0.

4.10.1.3 SYS$MSBDRIVER Removed from OpenVMS Distribution

V7.0

The driver for the Microsoft Windows Sound System ISA sound card (MSB), SYS$MSBDRIVER, has been removed from the OpenVMS Distribution as of Version 7.0. The following files have been removed:

- SYS$LOADABLE_IMAGES:SYS$MSBDRIVER.EXE
- SYS$EXAMPLES:SOUND_SERVICES.C
- SYS$EXAMPLES:SOUND_SAMPLE.C
- SYS$EXAMPLES:SOUND_SAMPLE.SND
- SYS$LIBRARY:SYS$STARLET_C.TLB module MSB.H

An enhanced version of this driver, called MMOV$MSBDRIVER, is included in Multimedia Services Version 1.4 for OpenVMS Alpha. This layered product also includes support for video capture and playback, an enhanced version of DECsound, and other audio and video applications.

MMOV$MSBDRIVER provides the same $QIO programming interface as SYS$MSBDRIVER. Digital recommends that the WAVE Applications Programming Interface provided by Multimedia Services for OpenVMS be used instead because it is more flexible and is portable to other platforms. (Multimedia Services Version 1.4 for OpenVMS is described in SPD 64.24.00.)

See Section 4.10.2.1 for information about a problem related to SYS$MSBDRIVER.

4.10.1.4 Enhanced SCSI Interface to IO$_DIAGNOSE Function

V7.0

The $QIO IO$_DIAGNOSE function has been enhanced to include 64-bit addressing support for the following SCSI class drivers: GKDRIVER, DKDRIVER, and MKDRIVER. For more information, see the *OpenVMS Alpha Guide to 64-Bit Addressing*.

4.10.1.5 PCI DMA Window Changes

V6.2

This release note should be read by anyone writing a driver for a PCI, EISA, or ISA device.

PCI bus implementations on Alpha platforms define two ways for PCI devices to access main memory: scatter/gather memory access and physical memory access. In **scatter/gather memory access**, the PCI address generated by a PCI device is translated to a main memory address by a scatter/gather table. In **physical memory access**, the PCI address generated by a PCI device is translated to a main memory address by the addition of a constant. Scatter/gather memory access is called **scatter/gather DMA**. Physical memory access is called **physical (or direct) DMA**.

Each memory access technique has advantages and disadvantages. Scatter/gather DMA allows access to all system memory, but it is more complex to program. Physical DMA is easier to program, but it may limit the amount of main memory that can be addressed.

Alpha PCI platforms implement scatter/gather DMA and physical DMA through **DMA address windows** on the PCI bus. A PCI DMA address window is an address range on the PCI bus through which PCI devices (and EISA/ISA devices behind the PCI/EISA bridge or PCI/ISA bridge) access main memory. The Alpha platforms that support PCI buses (AlphaServer 2000, AlphaServer 2100, AlphaServer 1000, AlphaServer 400, AlphaServer 8200, AlphaServer 8400, AlphaStation 200, AlphaStation 250) provide a minimum of two DMA address windows on the PCI bus. A DMA address window can be a physical DMA window, where a PCI bus address is a linear function of a system memory address, or a scatter/gather DMA window, where a PCI bus address undergoes a page table translation before becoming a main memory address.

Alpha platforms that support PCI buses have both a physical DMA window and a scatter/gather DMA window. In OpenVMS Alpha Version 6.2, the scatter/gather DMA window is based at PCI address 0 and extends to a maximum address of 3FFFFFFF (the actual size of the scatter/gather DMA window is a function of the amount of physical memory in the system). In OpenVMS Alpha Version 6.2, the physical DMA window is typically based at a PCI address above the scatter /gather DMA window.

To allow drivers to find the base of the physical DMA window in a platform-independent manner, two new function codes have been added to the bus support routine IOC$NODE_DATA. The basic definition of the IOC$NODE_DATA routine is as follows:

```
int ioc$node_data (CRB  *crb, int  function_code, void *user_buffer)
```

Inputs:

crb	Address of CRB. IOC$NODE_DATA uses the crb$l_node field and the vec$ps_adp field to find the data structures associated with the I/O bus to which this device is connected.
function_code	From [lib.lis]iocdef.sdl. Specifies information to be returned by IOC$NODE_DATA.
user_buffer	Address of caller's buffer. On success, the requested information is returned in the caller's buffer.

Outputs:

SS$_NORMAL	Normal successful completion. Requested information is returned in the caller's buffer.
SS$_ILLIOFUN	Unrecognized function code.
SS$_BADPARAM	Bad parameter. Usually this means that crb$l_node contains an invalid slot number. Check that the driver has been loaded with the /node qualifier.

When called with function code IOC$K_DIRECT_DMA_BASE, the IOC$NODE_DATA routine returns the base address of the physical DMA window. The base address is returned as a 64-bit value in anticipation of future 64-bit I/O buses. For this reason, the caller of the IOC$NODE_DATA routine should make sure the user_buffer argument points to a quadword cell when using the IOC$K_DIRECT_DMA_BASE function code.

When called with function code IOC$K_DIRECT_DMA_SIZE, the IOC$NODE_DATA routine returns the size of the physical DMA window, expressed in megabytes. The size of the direct DMA window is returned as a 32-bit value.

Using the physical DMA window for device DMA is straightforward. Once you have found the base of the physical DMA window using the IOC$NODE_DATA routine, you must adjust the DMA address that you assign to the device by adding the physical DMA window base to the main memory DMA buffer address. For example, on a typical system the physical DMA window is based at the PCI address 40000000 and extends to 7FFFFFFF. PCI addresses in the range from 40000000 to 7FFFFFFF are passed to main memory addresses 0 to 3FFFFFFF. This means that if you have a DMA buffer at main memory address 0, the PCI device would access this buffer at PCI address 40000000. The correspondence of main memory address and PCI DMA addresses is derived by the following formula:

```
PCI DMA address = (main memory buffer address) + (base of physical DMA window)
```

The main reason for locating the PCI DMA windows this way is to support bit-limited ISA devices. In OpenVMS Alpha Version 6.1, bit-limited ISA bus masters cannot use the scatter/gather DMA window because ISA bus masters do not have enough address bits to generate an address above 16 MB. For this reason, the scatter/gather DMA window was moved to PCI address 0. The scatter/gather table is now managed such that EISA and ISA devices that perform DMA in the scatter/gather window always get bus addresses below 16 MB.

Note that the size of the physical DMA window limits the amount of physical memory that can be addressed by a PCI, EISA, or ISA device. In the physical DMA window, it is not possible for an I/O device to address more than 1 GB of physical memory. For this reason, on large memory systems you may want to code your driver to perform DMA in the scatter/gather window. To perform scatter/gather DMA, use the standard counted resource management routines described in the Bookreader version of the *OpenVMS Alpha Device Support: Developer's Guide*.

4.10.1.6 Device IPL Setup for Drivers

V6.2 Alpha hardware platforms that support PCI, EISA, and ISA buses deliver I/O device interrupts at different IPLs, either 20 or 21. The IPL at which device interrupts are delivered can change if you move the device from one platform to another. This is a problem if the driver declares its device IPL to be 20, and then that driver is executed on a machine that delivers I/O device interrupts at IPL 21.

The simplest solution to this problem is for PCI, EISA, and ISA device drivers to use IPL 21. This works correctly on platforms that deliver I/O device interrupts at IPL 20 and on platforms that deliver I/O device interrupts at IPL 21.

A future release of OpenVMS Alpha may provide a platform-independent mechanism for drivers to determine the device IPL dynamically.

4.10.1.7 SCSI Port and Class Driver Changes

V6.2 The following sections describe changes to OpenVMS Alpha SCSI class and port device drivers.

4.10.1.7.1 OpenVMS Alpha SCSI Class Driver Changes

V6.2 Some SCSI-2 port interface (SPI) macros that supported OpenVMS Alpha SCSI class device drivers for Version 6.1 have been changed for OpenVMS Alpha Version 6.2. Table 4–2 lists the obsolete Version 6.1 macros and the Version 6.2 macros that have similar functions.

Note that many of the Version 6.2 SPI macros use different names, and most of them require new arguments. For this reason, do not replace each reference to an obsolete SPI macro without taking into account the OpenVMS Alpha Version 6.2 interface changes.

You can extract the new SPI$ interface macro definitions from SYS$LIBRARY:LIB.MLB, using the following command:

```
$ LIBRARY/EXTRACT=SPI$*/OUTPUT=SPI$DEFINITIONS.MAR SYS$LIBRARY:LIB.MLB
```

Table 4–2 Obsolete SPI Interface Macros

Obsolete V6.1 Macro	Similar Version 6.2 Macro
SPI$MAP_BUFFER	SPI$BUFFER_MAP
SPI$UNMAP_BUFFER	SPI$BUFFER_UNMAP
SPI$SEND_COMMAND	SPI$SEND_COMMAND
SPI$SET_CONNECTION_CHAR	SPI$CONNECTION_CHAR_SET
SPI$GET_CONNECTION_CHAR	SPI$CONNECTION_CHAR_GET
SPI$RESET	SPI$RESET_SCSI_BUS
SPI$ABORT_COMMAND	SPI$ABORT_COMMAND
SPI$DISCONNECT	SPI$DISCONNECT
SPI$ALLOCATE_COMMAND_BUFFER	SPI$CMD_BUFFER_ALLOC
SPI$DEALLOCATE_COMMAND_BUFFER	SPI$CMD_BUFFER_DEALLOC
SPI$RESERVED1	-
SPI$RESERVED2	-
SPI$CONNECT	SPI$CONNECT
SPI$FREEZE_QUEUE	SPI$QUEUE_FREEZE
SPI$TQE_WAIT	Use EXE$KP_TQE_WAIT
SPI$RELEASE_QUEUE	SPI$QUEUE_RELEASE
SPI$FLUSH_QUEUE	SPI$QUEUE_FLUSH
-	SPI$RESET_DEVICE

4.10.2 Problems and Restrictions

This section describes known problems and restrictions related to OpenVMS Alpha device support.

4.10.2.1 SYS$MSBDRIVER Load Failure

V7.0 On systems with ISA-only I/O buses, OpenVMS Alpha Version 7.0 attempts to load SYS$MSBDRIVER under the following conditions:

- The sound card is configured at the console with the ISACFG console command.
- Multimedia Services for OpenVMS is not installed.

- The user issues the SYSMAN IO command AUTOCONFIGURE.

This attempt fails because SYS$MSBDRIVER is no longer available with OpenVMS (see Section 4.10.1.3). This error does not affect the operation of the system, except that the sound card is not usable.

If the SYSMAN IO command AUTOCONFIGURE is issued on these systems, the following error messages are displayed:

```
%IOGEN-E-LOADERR, error configuring device AUA0 driver SYS$MSBDRIVER
-RMS-F-FNF, file not found
```

To determine if the sound card is configured, use the ISACFG -ALL console command, and look for device name "PCXBJ". If this device is configured, you can use the ISACFG -RM command to remove the device from the configuration.

This problem will be corrected in a future release of the operating system.

4.10.2.2 SCSI Firmware Support

The following sections relate to SCSI firmware support.

4.10.2.2.1 SCSI Firmware Update Instructions

V6.2 If your system contains RZ26L or RZ28 disks, the disks' firmware revision level must be at or above 442D for use in a SCSI VMScluster.

Important Note

Only certain RZ26L and RZ28 firmware revisions can be safely upgraded to firmware revision level 442D. Refer to Section 4.10.2.2.2 to determine if your disks are capable of being upgraded to firmware revision level 442D. If your disk is capable of supporting firmware revision level 442D, use the RZTOOLS Utility that is described in Section 4.10.2.2.3 to update the disk's firmware.

4.10.2.2.2 Firmware Revision Level 442D Requirements

V6.2 Only the following combinations of disk drives and firmware revision levels are capable of being upgraded safely to firmware revision level 442D. Performing the update procedure on any other combination can permanently damage the disk.

Table 4–3 Revision Level 442D Firmware Compatibility

Disk Drive	Firmware Revision	Disk File Name
RZ26L	440C	RZ26L_442D_DEC.FUP
RZ28	441C or D41C	RZ28_442D_DEC2104.FUP

4.10.2.2.3 442D Firmware Installation Procedure

V6.2 If you determine that your disk requires revision level 442D firmware, and it is capable of being upgraded safely, use the following procedure to update the firmware. (See Table 4–3 for the file name of the disk you are upgrading.)

```
$ MCR SYS$ETC:RZTOOLS_ALPHA DKB500 /LOAD=SYS$ETC:disk filename
  Read in 262144 bytes.
  Current FW version - X440C
  Upgrading to       - DEC0
  Loading code  ......
  New code has been sent to the drive.
```

4.10.2.3 SCSI Port and Class Drivers

V6.2 The following sections describe OpenVMS Alpha SCSI class and port device driver restrictions.

4.10.2.3.1 Mounting Third-Party SCSI Disks

V6.2 Users of the Digital RZ25M, Imperial MegaRAM MG-SCSI2, Seagate ST31200N, and ST31250N disks as well as other disks not qualified by Digital may see errors logged against these disks when they are mounted for the first time after booting in OpenVMS Alpha Version 6.2. Although these disks mounted error-free in OpenVMS Alpha Version 6.1, users might see 1 to 11 errors in OpenVMS Alpha Version 6.2 on the first mount.

The errors logged against these disks do not indicate a real problem. They occur because a 10-byte mode sense command is now issued to a disk instead of a 6-byte mode sense. This approach allows OpenVMS to work with a wider variety of third-party and other previously untried disks.

Digital is working with the parties responsible for maintaining the disk's firmware to correct this condition.

4.10.2.3.2 Add-On SCSI Adapters

V6.2 OpenVMS Alpha Version 6.2 supports various add-on SCSI adapters. Digital's AlphaGeneration platforms typically support one or more integral SCSI adapters, with the option of installing additional add-on SCSI adapters. Due to differences in device-naming conventions used between the Alpha console and OpenVMS, the OpenVMS device name may not match the name displayed by the console.

For example, the console designation for a SCSI device on the integral SCSI adapter may be DKA100. However, when two additional add-on SCSI adapters are added, the "A" designation becomes "C"; and DKA100 appears as DKC100 when OpenVMS is running.

Note that although the console and OpenVMS device names may be different, the unique specification of a device name from the console to the device name under OpenVMS will stay consistent, provided add-on SCSI adapters are not added or removed.

4.10.2.3.3 HSZ40 and Transportable SCSI Disk Shadow-Set Members

V6.2 An HSZ40 Raid-Array Controller provides the capability of an OpenVMS initialized SCSI disk (that is, one with a Files–11 ODS-2 format on it), to be moved between an OpenVMS controlled SCSI bus and an HSZ40 controlled SCSI bus, without reinitializing the disk and losing data. Disks that contain this functionality are called **transportable** disks.

A SCSI disk initialized by the HSZ40 and then subsequently initialized by OpenVMS is called a **nontransportable** disk, and cannot be moved to an OpenVMS controlled SCSI bus without losing data.

OpenVMS Volume Shadowing requires that a SCSI disk support the SCSI commands READ_LONG/WRITE_LONG. These SCSI commands in conjunction with OpenVMS Volume Shadowing are used to handle certain classes of errors as seen under normal volume shadowing operations. SCSI disks that support the READ_LONG/WRITE_LONG capability while connected to an OpenVMS controlled SCSI bus, lose this capability when the transportable disks are moved to the SCSI bus controlled by an HSZ40.

The lack of READ_LONG/WRITE_LONG capability is detected at shadow-set MOUNT time, by the following error:

```
MOUN$_DEVNOFE, device does not support FORCED ERROR handling
```

To correct this problem, specify the MOUNT qualifier /OVERRIDE=NO_FORCED_ERROR at shadow-set MOUNT time.

Note that specifying this MOUNT qualifier may cause shadow-set member SCSI disks to be removed from a shadow set if certain error conditions arise that cannot be corrected.

Digital recommends that HSZ40 nontransportable SCSI disks be used to contain shadow-set members that support READ_LONG/WRITE_LONG functionality, and offer benefits provided by the level of RAID chosen at initialization time.

4.10.2.3.4 SCSI Disk I/O Performance Degradation for KZMSA XMI and Adaptec 1742A Adapters

V6.2 As a result of the SCSI-2 Tagged Command Queuing (TCQ) support in OpenVMS Alpha Version 6.2, Digital has determined that customers with KZMSA XMI to SCSI and Adaptec 1742A adapters might experience a 20 percent SCSI disk I/O performance degradation because TCQ is not implemented for these adapters. The performance degradation is in the area of increased CPU cost per I/O. Customers running at less than maximum CPU utilization under OpenVMS Alpha Version 6.1 might not experience any degradation under OpenVMS Alpha Version 6.2.

Digital does not expect this situation to significantly affect DEC 7000 customers planning to upgrade to DEC 8000 Family systems using KZMSA adapters because the speed of those processors should offset the performance degradation. However, DEC 7000 customers who upgrade to OpenVMS Alpha Version 6.2 will experience the SCSI I/O disk performance degradation.

Digital expects that this will significantly affect existing DEC 2000 Model 300 systems customers that use the Adaptec 1742A SCSI adapter.

4.10.2.4 C Structure Member Names: Passing to Nested C Macros

V6.1

The file SYS$LIBRARY:SYS$LIB_C.TLB is a library of C header files that define structures corresponding to the structures defined for MACRO-32 in SYS$LIBRARY:LIB.MLB. These C structure definitions also include simple macros, which allow the use of the same field names in drivers written in C as those used in drivers written in MACRO-32 and BLISS.

A compile-time error results if both of the following occur:

- You pass one of these simple macro names for a structure member as part of a parameter to a C macro
- This macro passes this parameter to another macro

You can avoid this problem by using one of the following methods:

- "Flattening" the macro definition (by expanding the nested macros)
- Using C inline functions to replace the nested macros while maintaining the desired modularity
- Introducing additional temporary storage in the outer macro to avoid passing the input arguments directly to the nested macros

4.10.2.5 OpenVMS Alpha System-Code Debugger

V6.1

The OpenVMS Alpha System-Code Debugger has the following problems and limitations:

- The CALL command is not currently implemented.
- The SET WATCH command uses nonstatic watchpoints. Because nonstatic watchpoints are slower than static watchpoints, their use is not recommended.
- With the OpenVMS Debugger, if you use the SET BREAK/EXCEPTION command, when the program hits an exception, the user could fix the problem and continue the program. With the system-code debugger, the SET BREAK/EXCEPTION command catches the problem later (just before a bugcheck). Therefore, it is not possible to fix the problem and proceed. The GO command lets the system bugcheck and write the crash dump.
- Setting breakpoints at IF statements in the C programming language does not always work. For example, if you have code like the following and you set a breakpoint at the first line, the system-code debugger may or may not hit the breakpoint (depending on the condition):

  ```
  if (x)
  {
          ....
  } else
  {
            ...
  }
  ```

  To avoid this problem, you can set two breakpoints, one in the THEN clause and one in the ELSE clause, or set a breakpoint before the IF statement and then single-step until you reach the IF statement.

- When using the SET MODULE command on some images with the system-code debugger, you may receive an error message similar to the following:

  ```
  %DEBUG-E-INVPD, procedure descriptor at 00028840 is not valid.
  ```

This procedure descriptor is either in the init section for the image or in a paged-out image section. You can determine this by searching for the above value in the MAP file for the image to which the module belongs.

If the value is not within the init image section or a pageable section, file a Problem Report and provide the error message along with the MAP file for the image. ♦

4.11 Device Support on OpenVMS VAX Systems

VAX The following sections contain release notes pertaining to OpenVMS VAX device support.

4.11.1 Documentation Changes and Corrections

This section describes corrections to device support documentation.

4.11.1.1 *OpenVMS VAX Device Support Manual*

V6.1 The following sections describe corrections to the *OpenVMS VAX Device Support Manual*.

4.11.1.1.1 Linking a Device Driver

V6.1 Chapter 12 of the *OpenVMS VAX Device Support Manual*, Version 6.0, describes how to assemble, link, and load a device driver. In Step 3 of the procedure for preparing a driver for loading into the operating system, append the following text to the end of the procedure (following the paragraph that begins: “The resulting image must . . . ”):

To produce an image with a symbol table compatible with the System Dump Analyzer (SDA), you must link again; this time, using the UNIVERSAL=* option statement (to include all global symbols and to ensure proper state of the REL bits in the object records). Relink as shown in the following example:

```
$ LINK /NOEXECUTABLE/NOTRACEBACK/NOSYSSHR -
_$     /SYMBOLS=MYDRIVER.EXE,-
_$     /SHARE=DUMMY_FILE_NAME,-
_$     /NOMAP,MYDRIVER1.OBJ,MYDRIVER2.OBJ,-
_$     SYS.STB/SELECTIVE,-
_$     SYS$INPUT/OPTION
_$     BASE=0
_$     UNIVERSAL=*
```

For more information about the Linker, see the *OpenVMS Linker Utility Manual*.

4.11.1.1.2 Device-Register I/O Space: Usage Restrictions

V6.1 Chapter 5 of the *OpenVMS VAX Device Support Manual*, Version 6.0, describes device driver coding and the restrictions on the use of device-register I/O space. The third sentence of the fifth bulleted paragraph in Section 5.2 states that the instructions that refer to UNIBUS adapter registers must use longword context. This is the wrong bus. The sentence should read:

> “Instructions that refer to MASSBUS adapter registers must use the longword context.”

4.11.1.2 *OpenVMS VAX Device Support Reference Manual*

The following sections describe corrections to the *OpenVMS VAX Device Support Reference Manual*.

4.11.1.2.1 COM$DRVDEALMEM Routine Synchronization

V6.1

Chapter 3 of the *OpenVMS VAX Device Support Reference Manual*, Version 6.0, contains a section describing the COM$DRVDEALMEM routine.

At the end of the paragraph under *Synchronization*, add the following sentence:

"If called at IPL$_SYNCH or higher, the routine executes the fork process."

4.11.1.2.2 CRB Data Structure

V6.1

Chapter 1, Section 1.7, of the *OpenVMS VAX Device Support Reference Manual*, Version 6.0, contains Table 1-8 describing the CRB data structure fields. The description in the table for the CRB$L_INTD field is confusing and needs clarification. Replace the first two sentences in the CRB$L_INTD description as follows:

Field Name	Description
CRB$L_INTD	Portion of the interrupt transfer vector block that stores executable code, driver entry points, and I/O adapter information. This 10-longword area is overlaid with the contents of the interrupt transfer vector block that starts at VEC$L_INTD (offset 16) as described in Section 1.7.1. It contains pointers to the driver's . . .

4.11.1.2.3 SCDRP Data Structure SCSI Flags

V6.1

Chapter 1 of the *OpenVMS VAX Device Support Reference Manual*, Version 6.0, contains a section describing the SCDRP data structure SCSI flags.

In the SCDRP$L_SCSI_FLAGS field description for bit SCDRP$V_LOCK, make the following correction:

Change:	SCDRP$VLOCK
To:	SCDRP$V_LOCK

4.11.1.2.4 SPI$CONNECT Macro

V6.1

Chapter 2 of the *OpenVMS VAX Device Support Reference Manual*, Version 6.0, contains a section describing the SPI$CONNECT macro.

In the table listing the required inputs, add:

R4	Address of the SPDT

In the values returned in R3, the SPDT$M_CMDQ bit was added to the port capability mask (SPDT$L_PORT_FLAGS). When set, SPDT$M_CMDQ indicates that the port driver supports command queuing I/O.

In the return values table listing R3 and the mask bits (after SPDT$M_LUNS), add:

SPDT$M_CMDQ	Supports command queuing I/O

4.11.1.2.5 SPI$GET_CONNECTION_CHAR and SPI$SET_CONNECTION_CHAR Macros

V6.1 Chapter 2 of the *OpenVMS VAX Device Support Reference Manual*, Version 6.0, contains sections describing the SPI$GET_CONNECTION_CHAR and SPI$SET_CONNECTION_CHAR macros. Appended to the macro characteristics buffer is longword #12 for SCSI-2 support.

At the end of the characteristics buffer table in these macro descriptions, add the longword #12 information as follows:

12 SCSI-2 device characteristic status bits. Bits of this longword are defined as follows:

Bit	Description
0	When set, (SCDT$V_SCSI_2) indicates the device connection is SCSI-2 conformant.
1	When set, (SCDT$V_CMDQ) indicates the device connection supports command queuing.

4.11.1.2.6 $EQULST Macro

V6.1 Chapter 2 of the *OpenVMS VAX Device Support Reference Manual*, Version 6.0, contains a section describing the $EQULST macro.

In the parameter description for **symbol,value** insert the phrase *in decimal* as follows:

“ . . . and value specifies in decimal the value of the symbol.” ♦

4.12 Event Flags

This section contains release notes pertaining to event flags.

4.12.1 Changes and Enhancements

This section describes an enhancement to event flags.

4.12.1.1 Alternative to Local Event Flags

V7.0 With OpenVMS Version 7.0, event flags are divided into five clusters. The new special local cluster 4 supports only EFN 128. EFN 128 (symbolically represented as EFN$C_ENF) is intended for use with the wait forms of services (such as $QIOW or $ENQW) or with the $SYNCH system service. ENF$C_ENF need not be initialized, nor does it need to be reserved or freed. Multiple threads of execution can concurrently use EFN$C_ENF without interference. If EFN$C_ENF is used with system services such as $SETEF, $READ, and $CLREF, it performs as if it is always set. EFN$C_ENF can be used to eliminate the chance for event flag overlap. It can also be used when you do not care about the event flag, for example, when using $QIO with an AST completion.

For more information, see the revision of the *OpenVMS Programming Concepts Manual* that ships on the Version 7.0 CD–ROM. This information is also available in hard copy in the *OpenVMS Version 7.0 New Features Manual* under the title "New Local Event Flag EFN 128."

4.13 High-Performance Sort/Merge Utility (Alpha Only)

Alpha This section contains release notes pertaining to the SOR$ callable interface to the OpenVMS Alpha high-performance Sort/Merge utility. See the *OpenVMS Version 7.0 New Features Manual* for information about using this utility.

See Section 2.6.1 for release notes applicable to the high-performance Sort/Merge utility command line interface.

4.13.1 Problems and Restrictions

This section describes problems associated with the OpenVMS Alpha Version 7.0 release of the high-performance Sort/Merge utility.

4.13.1.1 Default File Specification

V7.0 The high-performance Sort/Merge utility does not take its default output file extension from the first input file (in instances where there is a user-specified input file), unlike the Sort/Merge utility.

To avoid this situation, you should explicitly specify the output file extension. This difference in behavior will be fixed in a future major release.

4.13.1.2 Error Messages

V7.0 The following problem pertains to error messages generated by the high-performance Sort/Merge utility. Error messages that would normally refer to one or more files by name, instead contain the FAO directive that would be used to generate the file name. In addition, error messages do not include RMS or other secondary error messages. For example, if the high-performance Sort/Merge utility cannot find the X.DAT input file, it generates the message:

```
%SORT-F-OPENIN, error opening !AS as input
```

The Sort/Merge utility generates a message that contains the file name:

```
%SORT-F-OPENIN, error opening DISK$2:[SORT_AREA]X.DAT; as input
-RMS-E-FNF, file not found
```

This difference in behavior will be fixed in a future major release.

4.13.1.3 Concurrent Sort Operations

V7.0 Memory allocation differences may limit the high-performance Sort/Merge utility's ability to perform the same number of concurrent sort operations as can be performed by the Sort/Merge utility in the same amount of virtual memory.

If this situation occurs, you can either increase the amount of virtual memory that is available to the process, or reduce the working set extent (WS_EXTENT).

4.13.1.4 Merging Stream Files Limitation

V7.0 A limitation exists for merging stream files in this release of the high-performance Sort/Merge utility. The end of file is not written correctly for the output stream file. To work around this limitation, explicitly specify the format of the output file as fixed or variable. This can be done at the application programming interface using the following:

```
    sor$pass_files(...,rfm,...)
```

where "rfm" represents the record format of the output file as either FAB$C_FIX or FAB$C_VAR.

If you want the output file in stream format, the RMS Convert utility can be used to convert the output file from fixed or variable record format to stream format.

This limitation will be removed in a future major release. ♦

4.14 I/O User

The following sections contain release notes pertinent to the I/O user.

4.14.1 Problems and Restrictions

This section describes known problems with I/O write operations.

4.14.1.1 I/O Write Operations Interrupted by System Failure

V6.0

The OpenVMS operating system ensures that when an I/O write operation returns a successful completion status, the data is available on the disk or tape media.

OpenVMS guarantees atomicity for single-block I/O write operations to DIGITAL Storage Architecture (DSA) drives.[1] However, if a system failure occurs while a multiple block I/O write operation is in progress, the OpenVMS operating system does not guarantee the successful completion of the I/O write. When a failure interrupts a write operation, the data can be left in any one of the following conditions:

- The new data is written completely to the disk blocks on the media, but a completion status was not returned before the failure.
- The new data is partially written to the media so that some of the disk blocks involved in the I/O contain the data from the in-progress I/O write and other blocks contain the data that was present prior to the I/O write operation. Note that blocks updated with new data can be interspersed with blocks containing old data.
- The new data was never written to any of the disk blocks on the media.

Individual blocks can contain either all new data or all old data; a block is not partially updated when dealing with DSA devices.

To guarantee that an I/O write operation either completes successfully, or (in the event of failure) is redone or rolled back as if it were never started, applications must be supplemented by additional techniques to ensure data correctness and recovery. For example, using RMS Journaling for OpenVMS or database or file journaling and recovery techniques allows applications to automatically recover from failures such as:

- Permanent loss of the path between a CPU data buffer containing the data being written and the disk being written to during a multiple block I/O operation. Communication path loss can occur due to node failure or a failure of node-to-node communications. Note that transient failures are automatically recovered from during mount verification without requiring any further recovery techniques.
- Failure of a CPU (such as a system crash, halt, power failure) during a multiple block I/O write operation.
- Mistaken deletion of a file.
- OpenVMS RMS incomplete bucket write operation to an indexed file.

[1] Single block atomicity may not be guaranteed with SCSI devices. This includes the new HSZ and HSJ SCSI adapters that extend SCSI disk connectivity to the HSC family of devices. An OpenVMS operating system makes no distinction between the SCSI-adapted DSA devices and actual SCSI devices.

- Cancellation of an in-progress multiple block write operation.

Note that Volume Shadowing for OpenVMS promotes availability by providing access to data in the presence of media deterioration, communication path failure, or controller or device failure. However, volume shadowing should not be mistaken for a journaling product that tracks changes to files for data integrity.

4.14.2 Documentation Changes and Corrections

This section describes changes to I/O documentation.

4.14.2.1 *OpenVMS I/O User's Reference Manual*

V6.1

The following section should appear in the Terminal Driver chapter of the *OpenVMS I/O User's Reference Manual*.

Terminal Devices Supported

Table 4–4 lists the supported terminal devices for the Digital 2100 Server Model A500/600 MP and the DEC 2000 Model 300.

Table 4–4 Supported Terminal Devices

Terminal Interface	Number of Lines	Output		Split Speed	Bus	International Modem Control
		Silo	DMA			
Digital 2100 Server Model A500/600MP	2	No	No	No	None	Full
Digital 2100 Server Model A500/600MP[1]	4, 8	Yes	No	No	EISA	Full
DEC 2000 Model 300	2	No	No	No	None	Full
DEC 2000 Model 300[1]	4, 8	Yes	No	No	EISA	Full

[1] Optional multiplex serial DIGIBOARD PC/X™ adapter card. You can daisy chain up to 4 boards in one system, resulting in 16, 32, or 64 ports.

4.15 Librarian Utility

The following sections contain release notes pertaining to the Librarian utility (LIBRARIAN).

4.15.1 Problems and Restrictions

This section describes known LIBRARIAN problems and restrictions.

4.15.1.1 PGFLQUOTA Should Exceed 23000 (Alpha Only)

V1.5

Alpha

The OpenVMS Alpha LIBRARIAN sometimes does not inform you of errors during compression, data reduction, or data expansion operations. This problem occurs if the account or process in which the LIBRARIAN is running has a low PGFLQUOTA process quota. Operation failure is not readily apparent because the $PUTMSG system service always returns a status of SS$_NORMAL, even when the system service fails. However, when a failure occurs, the LIBRARIAN returns a status other than success.

To work around this problem, run the compression, data reduction, or data expansion operation in an account with a PGFLQUOTA process quota greater than 23000. In addition, ensure that your command procedures check the return status from the LIBRARY command. ♦

4.16 Linker Utility

The following sections contain release notes pertaining to the OpenVMS Alpha Linker utility (linker).

4.16.1 Problems and Restrictions

This section describes known linker problems and restrictions.

4.16.1.1 Fixup Information Not Generated (Alpha Only)

V1.0

Alpha

The linker has been modified so that a new error message informs you at link time that global symbols from shareable images are being placed into byte- or word-sized fields by the linker. (Word- and byte-sized stores of global symbols do not generate fixup information. Fixup information is required when linking against shareable images.) When this situation occurs, an error message is printed, and image production is inhibited.

The following example shows this new error message:

```
%LINK-E-NOFIXSYM, unable to perform WORD fixup for symbol TPU$_OPTIONS
        in psect $PLIT$ in module TEST_MODULE file USER:[ACCOUNT]TEST.OLB;1
```

To work around this restriction, move the symbolic value into the desired location at run time rather than at link time. For example, in MACRO, rather than performing .WORD TPU$_OPTIONS, use the instruction MOVW #TPU$_OPTIONS,*dest*. ♦

4.16.1.2 Shareable Image Psects Pointed to by Symbol Definitions (Alpha Only)

V1.0

Alpha

The linker cannot overlay program sections that are referenced by symbol definitions with shareable image program sections of the same name. Symbol definition records that contain the index of an overlaid program section are generated by the C compiler when the relaxed ref-def extern model is used (the default). Shareable image program sections are created when you link a shareable image and use the PSECT keyword in your SYMBOL_VECTOR option.

If the linker detects this condition, it issues the following error:

```
%LINK-E-SHRSYMFND, shareable image psect <name> was pointed to by a symbol definition
%LINK-E-NOIMGFIL, image file not created
```

The link continues, but no image is created. To work around this restriction, change the symbol vector keyword to DATA, or recompile your C program with the qualifier /EXTERN=COMMON. ♦

4.16.2 Documentation Changes and Corrections

This section describes corrections to linker documentation.

4.16.2.1 *OpenVMS Linker Utility Manual*

V1.5

The *OpenVMS Linker Utility Manual* incorrectly describes /NONATIVE_ONLY to be the default behavior of the linker.

The /NATIVE_ONLY qualifier to the LINK command directs the linker *not* to pass along the procedure signature block (PSB) information, created by the compilers, in the image it is creating. This is the default behavior of the linker. For more information, refer to *Migrating an Application from OpenVMS VAX to OpenVMS Alpha*.

4.17 Lock Manager

The following sections contain release notes pertaining to the lock manager.

4.17.1 Changes and Enhancements

This section describes changes or enhancements to the software.

4.17.1.1 LCKMGR Spin Lock (Alpha Only)

V6.1

Alpha

The synchronization for the OpenVMS lock manager has changed with this release of OpenVMS Alpha. A new spin lock with a name of LCKMGR is now used to synchronize the OpenVMS lock manager for standalone OpenVMS Alpha machines. Alpha systems running as part of a VMScluster still synchronize with the SCS (IOLOCK8) spin lock. Use of the LCKMGR spin lock in a VMScluster system correctly locks the SCS spin lock.

This change has no impact for users of the system service interfaces of $ENQ[W], $DEQ, and $GETLKI[W].

If you have code that currently accesses OpenVMS internal lock manager data structures such as LKBs or RSBs, you must modify your software to correctly synchronize with the lock manager by using the LCKMGR spin lock. This change does not affect OpenVMS VAX software. ♦

4.18 LTDRIVER

The following sections contain release notes pertaining to the LTDRIVER.

4.18.1 Problems and Restrictions

This section describes known LTDRIVER problems and restrictions.

4.18.1.1 CANCEL SELECTIVE Cannot Cancel IO$_TTY_PORT Functions

V6.1

In prior releases, LTDRIVER did not set the "extended DDT" bit; therefore, the POSIX function CANCEL SELECTIVE did not work with LTDRIVER. This has been fixed, but a restriction remains.

Although this fix allows $QIO reads and writes to be selectively canceled, any $QIO done to the port driver (that is, with the IO$_TTY_PORT function modifier—like a LAT connect $QIO) *cannot* be canceled with CANCEL SELECTIVE. This problem will be addressed in a future OpenVMS release.

4.19 MACRO-32 Compiler for OpenVMS Alpha (Alpha Only)

Alpha

The following sections contain information pertaining to the MACRO-32 Compiler.

4.19.1 Problems and Restrictions

This section describes problems and restrictions pertaining to the MACRO-32 Compiler for OpenVMS Alpha.

4.19.1.1 .CALL_ENTRY Requires New STARLET.MLB

V7.0

The MACRO-32 compiler has been modified to include 64-bit addressing support. When you use this new version of the compiler, regardless of whether you use the 64-bit addressing features, you must also use the OpenVMS Version 7.0 ALPHA$LIBRARY:STARLET.MLB.

STARLET.MLB contains the MACRO_COMPILER_DIRECTIVES macro definitions, including the revised .CALL_ENTRY directive. If you do not use the OpenVMS Version 7.0 ALPHA$LIBRARY:STARLET.MLB, your program will not compile successfully. Instead, the following errors will be reported:

```
%AMAC-E-DIRARGERR, error in compiler directive argument
at line number n in file filename
```

```
%AMAC-E-DIRSYNX, directive syntax error
at line number n in file filename
```

Make sure that the OpenVMS Version 7.0 ALPHA$LIBRARY:STARLET.MLB is on your system and that the ALPHA$LIBRARY logical points to the correct directory.

If necessary, you can extract .CALL_ENTRY from the new STARLET.MLB, and insert it in an alternate STARLET, if you are using one.

4.19.1.2 .ASCID Directive

V6.2

The length of the argument to a .ASCID directive is limited to 996 characters when using the OpenVMS Alpha MACRO-32 Compiler. No such restriction exists in the VAX MACRO Assembler.

This newly discovered difference will remain a restriction of the compiler.

4.19.1.3 .GLOBAL_LABEL Directive

V6.2

Labels declared with the .GLOBAL_LABEL directive can be used as the **newpc** argument in calls to the $UNWIND (Unwind Call Stack) system service because it allows the address of the label to be stored.

However, there is no provision in the compiler to automatically adjust the stack pointer at such labels to remove arguments passed on the stack or compensate for stack alignment. If the call stack is unwound back to an alternate PC in the calling routine, the stack may still contain arguments and alignment bytes, and any stack-based references that expect this adjustment to the caller's original stack depth (which happened automatically on VAX) will be incorrect.

Code that contains labels declared with this directive that are to be used as alternate PC targets for $UNWIND must be examined carefully to ensure correct behavior, with particular emphasis on any references based on the stack pointer.

4.19.1.4 Floating-Point Return Values in R0

V6.1

A MACRO program that calls out to a routine and expects a floating-point return value in R0 may require a "jacket" between the call and the called routines to move the returned value from floating-point register 0 to R0.

4.19.1.5 INSV Instructions Negate Granularity Preservation

V6.1 INSV instructions do not generate code that correctly preserves granularity when granularity preservation is turned on.

4.19.1.6 .JSB_ENTRY Directive

V6.1 For procedures declared with the .JSB_ENTRY directive, the MACRO-32 Compiler automatically generates a null frame procedure descriptor, independent of debug or optimization qualifiers. The null frame procedure descriptor allows for debugging of problems with the linkage itself.

Because no new context is set up by a null frame procedure, a side effect is that there is no guarantee of completely accurate debugger information about such procedures in response to SHOW CALLS and SHOW STACK commands. For example, the line number in the called null procedure (to which a JSB is done) may be reported as the line number in the calling procedure from which the JSB is issued.

4.19.1.7 Floating-Point and Packed-Decimal Instruction Arguments

V1.5 Because packed-decimal instructions and floating-point instructions are implemented by means of macros, there is one restriction on the format of the arguments. In a macro invocation, an initial circumflex (^) is interpreted to mean that the parameter is a string and the character immediately following the circumflex is the string delimiter. Because of this, you cannot use arguments that begin with an operand type specification, such as ^x20(SP). Note that immediate mode arguments, such as #^XFF, can use an operand type specification because the circumflex is not the initial character.

4.19.1.8 Multiple Object Files Compilation

V1.5 The MACRO-32 Compiler does not support the creation of separate object files from multiple source files specified in one command. It does support the creation of one object file from multiple source files specified in one command.

4.19.2 Documentation Changes and Corrections

This section describes changes in the MACRO–32 compiler's documentation.

4.19.2.1 *Porting VAX MACRO Code from OpenVMS VAX to OpenVMS Alpha*

The following sections describe corrections to *Porting VAX MACRO Code from OpenVMS VAX to OpenVMS Alpha*.

4.19.2.1.1 Maintaining Common Sources for VAX and Alpha

V7.0 Section 1.6.1 of *Porting VAX MACRO Code from OpenVMS VAX to OpenVMS Alpha* refers to an obsolete file, SYS$SHARE:MACRO_COMPILER_DIRECTIVES.MAR. Starting with OpenVMS Version 6.1, that file is no longer separately available on the kit. Instead, its contents are included in SYS$LIBRARY:STARLET.MLB.

If you added any compiler directives to your code, they will be resolved by the library SYS$LIBRARY:STARLET.MLB when the code is assembled. The assembler automatically searches this library for any undefined macros. After finding these definitions, the assembler ignores occurrences of the compiler directives.

However, if you are using OpenVMS VAX Version 6.0 or earlier, you must first extract the directive definitions from ALPHA$LIBRARY:STARLET.MLB on Alpha and insert them into your SYS$LIBRARY:STARLET.MLB on OpenVMS VAX. For example:

```
LIB/EXTRACT=.JSB_ENTRY/OUT=JSB_ENTRY.MAR ALPHA$LIBRARY:STARLET.MLB
.
.
.
LIB/INSERT SYS$LIBRARY:STARLET.MLB JSB_ENTRY.MAR
```

Note that many of the definitions of the compiler directives refer to other macros. Be sure to extract not only the definitions of all the compiler directives used in your code but also all the associated macros; then insert them all into SYS$LIBRARY:STARLET.MLB on your OpenVMS VAX system.

4.19.2.1.2 Packed-Decimal and Floating-Point Instructions and the VAX AP

V7.0 Section 2.8.1 of *Porting VAX MACRO Code from OpenVMS VAX to OpenVMS Alpha* contains an unnumbered list item that reads as follows:

> Use of argument registers (R16 through R21)
>
> Arguments are passed to the emulation routine by means of the argument registers (R16 through R21); attempts to use these registers as arguments in a packed decimal instruction might not work correctly.

The last phrase “might not work correctly” should be amended to read “will not not work correctly.” The same is true for floating-point instructions (see Section 2.9.1 of the same manual).

The compiler converts references to the VAX argument pointer (AP) into references to the Alpha argument registers (see Section 2.3 of the same manual). For example, the compiler converts a parameter reference off the VAX AP, such as 4(AP), to Alpha R16.

To prevent these problems with the AP-based references when packed decimal or floating-point instructions are used, it is simplest to specify /HOME_ARGS=TRUE on the entry point of the routine that contains these references. This forces all AP-based references to be read from the procedure frame, rather than the argument registers, so that no changes need to be made to the instructions.

If /HOME_ARGS=TRUE is not used, the source code that contains parameter references based on the AP register as operands to packed-decimal instructions (or floating-point instructions) should be changed. Copy any AP-based operand to a temporary register first, and use that temporary register in the packed-decimal or floating-point instruction. For example, change the following code:

```
MOVP    R0, @8(AP), @4(AP)
```

to:

```
MOVL    8(AP), R1
MOVL    4(AP), R2
MOVP    R0,(R1),(R2)
```

4.19.2.1.3 Alignment Considerations for Atomicity: Correction to Example

V7.0 The example shown in Section 2.10.5 of *Porting VAX MACRO Code from OpenVMS VAX to OpenVMS Alpha* incorrectly shows the EXTWL, INSWL, and MSKWL instructions operating on R28. Instead, the example should show these instructions operating on R1.

The example with corrections follows:

```
        BIC     R1,#^B0110,R28  ; Compute Aligned Address
Retry:  LDQ_L   R24,(R28)       ; Load the QW with the data
        EXTWL   R24,R1,R23      ; Extract out the Word
        ADDL    R23,#1,R23      ; Increment the Word
        INSWL   R23,R1,R23      ; Correctly position the Word
        MSKWL   R24,R1,R24      ; Zero the spot for the Word
        BIS     R23,R24,R23     ; Combine Original and New word
        STQ_C   R23,(R28)       ; Conditionally store result
        BEQ     fail            ; Branch ahead on failure
        .
        .
        .
Fail:   BR      Retry
```

4.19.2.1.4 Entry Point Declarations

V6.1 Any code label that is a possible target of a CALLS, CALLG, JSB, BSBW, or BSBB instruction must be declared as an entry point. In addition, any code label must be declared as an entry point using a .JSB_ENTRY or .JSB32_ENTRY directive if:

- The label can be the target of a global (cross-module) JMP, BRB, or BRW instruction
- The label can be the target of an indeterminate branch (such as BRB @(R10)), where the address of the label has been stored in R10, even if the reference and the label are within the same module
- The address of the label is stored in a register or memory location, even if it is never accessed by the current module

The OpenVMS calling standard for Alpha computers does not provide a way to access indeterminate code addresses directly. All such accesses are accomplished using a procedure descriptor to describe the routine and the code address. When a code label address is stored, the compiler does not know if that address will be referenced only by the current module, or whether it may be accessed by another MACRO module or another module written in another language. Whenever the MACRO–32 compiler stores a code address, it always stores the procedure descriptor for that address so that other code can access it correctly. For a procedure descriptor address to exist, the label must be declared as an entry point.

Likewise, when a stored address is used as a destination, the compiler does not know where that address came from, so it always assumes that the stored address is the address of a procedure descriptor and uses that descriptor to pass control to the routine.

4.19.2.1.5 BUG*x* Instruction Is Not Supported

V1.5 The BUG*x* instruction is not supported by the compiler. This instruction was inadvertently omitted from the list of unsupported instructions in *Porting VAX MACRO Code from OpenVMS VAX to OpenVMS Alpha*.

4.19.2.1.6 /FLAGGING=INSTRUCTIONS Qualifier

V1.5

When /FLAGGING=INSTRUCTIONS is enabled, absolute addresses detected by the compiler are flagged. For example, MOVL R0, 200 compiles correctly (updating memory location 200), but the desired absolute address may be different on an Alpha computer. As a result, the informational message CHKABSADR is reported. This does not also apply to the instruction M*x*PR, as stated in the description of the /FLAGGING=INSTRUCTIONS qualifier in Appendix A.

If you do not want these information messages reported, specify the /NOFLAGGING=INSTRUCTIONS qualifier.

4.19.2.1.7 TRACEBACK and DEBUG Options Restriction Removed

V1.5

The restriction that pertained to the qualifiers /DEBUG, /DISABLE, and /ENABLE and to the directives .DISABLE and .ENABLE, which is documented in Appendixes A and B, has been removed.

Contrary to the documentation, you can initially enable the TRACEBACK and DEBUG options with the /ENABLE qualifier. You can then turn the options off and on with the .DISABLE and .ENABLE directives for whichever code sections you want. However, if the /DEBUG qualifier is used in the command line, it overrides /ENABLE=(DEBUG,TRACEBACK) and /DISABLE=(DEBUG,TRACEBACK), regardless of their position on the command line. ♦

4.20 Mathematics (MTH$) Run-Time Library

The following sections contain release notes pertaining to the Run-Time Mathematics Library (MTH$).

4.20.1 Problems and Restrictions

This section describes known MTH$ problems and restrictions.

4.20.1.1 Linking Images to Run on Previous OpenVMS VAX Versions (VAX Only)

V6.1

VAX

This version of OpenVMS VAX provides updated versions of the Mathematics Run-Time Library (RTL) images MTHRTL.EXE, UVMTHRTL.EXE, and VMTHRTL.EXE that contain new entry points in support of DEC Fortran Version 6.0. (UVMTHRTL.EXE is an alternate form of MTHRTL.EXE; references to MTHRTL.EXE in the following paragraphs also apply to UVMTHRTL.EXE.)

Due to the large number of entry points added to MTHRTL.EXE, that image's transfer vector was extended and its global section match identifier incremented. This means that images linked against the new version of MTHRTL.EXE will not run on a system running a previous version of OpenVMS VAX, unless that system has also installed DEC Fortran Version 6.0. In addition, images linked against the new MTHRTL.EXE cannot be translated to run on OpenVMS Alpha using DECmigrate.

To link an image so that it will run on a previous version of OpenVMS VAX, create a directory that contains saved copies of the .EXE and .OLB files from the SYS$LIBRARY directory of the earliest version you wish to support, and define the logical name SYS$LIBRARY to point to that directory before linking. Because OpenVMS VAX also defines a system logical name MTHRTL to refer to either MTHRTL.EXE or UVMTHRTL.EXE, you must also define MTHRTL as a logical name in the process or job table to point to the copy in the directory of older images. For example:

```
$ DEFINE/USER SYS$LIBRARY disk:[OLD_SYSLIB]
$ DEFINE/USER MTHRTL SYS$LIBRARY:MTHRTL.EXE
$ LINK ...
```

Images to be translated using DECmigrate should be linked against the SYS$LIBRARY files of OpenVMS VAX Version 5.5-2 or earlier. ♦

4.21 Migration (VAX to Alpha)

The notes in this section contain information of interest to users who are migrating from an OpenVMS VAX system to an OpenVMS Alpha system.

4.21.1 Documentation Changes and Corrections

The following note describes changes in the migration documentation.

4.21.1.1 New Manuals Replace Old Ones

V7.0 These old migration manuals have been retired:

- *Migrating to an OpenVMS AXP System: Planning for Migration*
- *Migrating to an OpenVMS AXP System: Recompiling and Relinking Applications*

Much of the information that was contained in these manuals is included in two new migration manuals that ship with OpenVMS Version 7.0:

- *Migrating an Application from OpenVMS VAX to OpenVMS Alpha*
- *Migrating an Environment from OpenVMS VAX to OpenVMS Alpha*

The new manuals also include information from two other retired manuals:

- *OpenVMS Compatibility Between VAX and Alpha*
- *A Comparison of System Management on OpenVMS AXP and OpenVMS VAX*

4.22 POLYCENTER Software Installation Utility

The notes in this section pertain to the POLYCENTER Software Installation utility. Also see Section 3.28 for notes about this utility that are of interest to system managers.

4.22.1 Problems and Restrictions

This section describes known problems and restrictions with using the POLYCENTER Software Installation utility to create software kits. Problems and restrictions of interest to system managers are described in Section 3.28.1.

4.22.1.1 Account Removal Problem

V6.1 Removing a product with the POLYCENTER Software Installation utility results in the removal of accounts specified by *account* statements. The utility should not do this in all cases because the file SYSUAF.DAT may be clusterwide or local to a single system disk or groups of system disks.

If you install software on a particular system disk, any *account* statements create the necessary accounts. If you install the software on a second system disk that shares the same SYSUAF.DAT file, the account already exists and does not need to be created.

When you remove a product, the utility always removes the account, regardless of whether the SYSUAF.DAT file is shared by another system disk.

The same problem exists with the *rights identifier* statement and the file RIGHTSLIST.DAT.

Digital expects to correct these problems in a future release.

4.22.1.2 Alternate File Placement Not Supported

V6.1 You cannot specify an alternate location for some of your product files using the POLYCENTER Software Installation utility.

Digital expects to correct this problem in a future release.

4.22.1.3 Conflict Error Reporting

V6.1 Some error messages indicating managed object conflicts do not identify the related products.

4.22.1.4 Error Reporting

V6.1 When packaging a kit, you may receive many different errors (for example, wrong keywords or missing directives) in your product description file (PDF). In several cases, the utility error messages do not contain enough information to determine the problem with the PDF.

Digital expects to correct this problem in a future release.

4.22.1.5 File Generation Restrictions

V6.1 The generation option to the *file* statement does not work correctly under some circumstances.

For example:

```
-- PDF #1:
product DEC VAXVMS TEST1 V1.0 full ;
file [SYSMGR]TEST.EXE generation 1 ;
end product ;

-- PDF #2:
product DEC VAXVMS TEST2 V1.0 full ;
file [SYSMGR]TEST.EXE generation 2 ;
end product ;
```

Installing TEST1 then TEST2 works correctly. However, if you remove TEST2, generation one of the TEST.EXE file is not reinstalled and the utility displays a message about replacement material being unavailable.

One workaround is to have TEST2 use the *execute install* statement to execute a procedure that saves previous versions of TEST.EXE. A corresponding *execute remove* could restore it.

Another related problem is that you can remove TEST1 and then TEST2, but only in that order. The previous example shows removing TEST2 first which is the logical order (since it is the reverse order of installation).

Note that if you install TEST2 then TEST1, the installation of TEST1 generates an error message about replacement material being unavailable. The utility should give a better error message that indicates that a newer generation of the file is already installed on the system.

Digital expects to correct these problems in a future release.

4.22.1.6 Information Statement Problem

V6.1

Product Description File (PDF) *information* statements that you want to display to the user during the configuration phase are not displayed if the user enters Yes (the default answer) to the following question:

```
Do you want all the default values for this product? [YES]
```

Digital expects to correct this problem in a future release.

4.22.1.7 Multiple Execute Remove Statements

V6.1

There is a problem with the *execute remove* statement where only the first one executes during a remove operation. However, all of the *execute install* statements execute during an install operation.

Digital expects to correct this problem in a future release.

4.22.1.8 Multiple Operating Scopes Not Supported

V6.1

Multiple operating scopes are not supported and may not work correctly in certain circumstances.

4.22.1.9 Package Operation Constraints

V6.1

A typical task when creating software kits is to create a sequential kit by performing a package operation. If you use the same directory for the source and destination for your package operation, you can create problems for future install operations that use the directory as a source.

If both a sequential kit (with file type .PCSI) and a product description file for a reference kit (with file type .PCSI$DESCRIPTION) are in the same directory, an install operation that points to that source area always opens the file with file type .PCSI$DESCRIPTION. This can cause errors locating product material if it is not in the same place. There is no way to specify the sequential kit for the installation when a reference kit for the same product is present.

To avoid this problem, use separate directories for the source and destination of your package operations.

4.22.1.10 Partial Kit Restrictions

V6.1

Kits of type partial do work correctly under many circumstances. Digital recommends that you avoid creating partial kits with this version of the POLYCENTER Software Installation utility because of the following problems and restrictions:

- The utility does not require that a full kit of the product has already been installed.
- If a full product kit is not found, the utility attempts to attach the partial kit to the operating system as its parent.
- When removing a partial kit that has the operating system as its parent, the utility also removes the operating system without displaying any messages.

If possible, use patch or mandatory update kit types rather than partial kits.

Digital expects to correct these problems in a future release.

4.22.1.11 Rights Identifier Problem

V6.1 Refer to Section 4.22.1.1 for a description of this problem.

4.22.1.12 “Uses” Clause File Specification Restriction

V6.1 You cannot specify the same file in the “uses” clause of two separate execute statements in the same product description file (PDF), for example:

```
product DEC AXPVMS MYSTUFF full;
     ...
     execute install "@doit thisway" uses [000000]doit.com;
     execute install "@doit thatway" uses [000000]doit.com;
     ...
end product;
```

The file is deleted before the second use.

Digital expects to correct this problem in a future release.

4.22.2 Documentation Changes and Corrections

This section describes changes in the utility’s documentation.

4.22.2.1 DCL Help: PRODUCT PACKAGE Command

V6.2 The description of the PRODUCT PACKAGE command and its qualifiers has been significantly enhanced. To view these changes in DCL Help, enter the HELP PRODUCT PACKAGE command at the DCL prompt.

4.22.2.2 *POLYCENTER Software Installation Utility Developer’s Guide*

The following sections describe corrections to the *POLYCENTER Software Installation Utility Developer’s Guide*. These corrections are included in the versions of this manual that ship on the Version 7.0 CD–ROM.

4.22.2.2.1 File Statement

V6.2 In the *POLYCENTER Software Installation Utility Developer’s Guide*, the **file** statement is described in the Product Description Language (PDL) reference section. Replace the description of the **[no]generation** option as follows:

[no]generation *generation*
Specifies the generation number of the file as an unsigned integer from 0 to 4294967295. Using the **[no]generation** clause is equivalent to specifying a value of 0; both mean the file has no explicit generation number. The default for the file statement is **nogeneration**. Refer to the Description section for the meaning of this value.

4.22.2.2.2 Scope Statement

V6.1 In the *POLYCENTER Software Installation Utility Developer’s Guide*, the **scope** statement is described in the Product Description Language (PDL) reference section and in Appendix B. The following notes apply to the scope statement:

- *Bootstrap Scope* should be used for products that use device drivers, especially those drivers that must be read by the primitive file system. Because files in Bootstrap Scope are read by the primitive file system, they are read when not synchronized with the file system on other cluster members that might access the same disk. Therefore, those files must retain stable positions as long as the disk is in use from *any* system and must not be manipulated by online disk defragmentation operations, even those that use the MOVEFILE primitive.

- *Product Scope* is the default scope for most objects; therefore, it is generally unnecessary to specify Product Scope.
- *Operating Scope* is not implemented for this version of the POLYCENTER Software Installation utility.

4.23 Privileged Interfaces and Data Structures (Alpha Only)

V7.0

Alpha

OpenVMS Alpha Version 7.0 includes significant changes to OpenVMS Alpha privileged interfaces and data structures. As a result of these changes, some customer-written device drivers and applications with inner-mode code may require source changes.

For more details about changes to OpenVMS Alpha privileged interfaces and data structures, see the *OpenVMS Alpha Guide to Upgrading Privileged-Code Applications*. ♦

4.24 Run-Time Libraries

The following sections contain release notes pertaining to run-time libraries.

4.24.1 Changes and Enhancements

The following note describes a change in status of the PPL.

4.24.1.1 PPL (Parallel Processing Library) to Be Retired

V7.0

The OpenVMS Version 7.0 operating system includes the PPL (Parallel Processing Library) run-time library in maintenance mode only. PPL provides routines that assist programmers in developing concurrent programs using multiple processes.

The PPL library will be retired in a future release, but will still be provided on the freeware CD–ROM that ships with the OpenVMS operating system.

If you have used PPL, consider whether you can use the OpenVMS DECthreads library to implement concurrent applications. The DECthreads library has shipped with the OpenVMS VAX operating system since Version 5.5 and with the OpenVMS Alpha operating system since Version 1.0. Starting with OpenVMS Version 7.0, DECthreads supports multiprocessing as part of its multithreading capabilities. In addition, DECthreads implements the industry-standard POSIX multithreading interface. For more information about DECthreads, see the *Guide to DECthreads*.

4.24.2 Problems and Restrictions

This section describes known run-time library problems and restrictions.

4.24.2.1 Run-Time Libraries Not Included in OpenVMS Alpha Systems (Alpha Only)

V1.5

Alpha

The run-time libraries listed in Table 4–5 are not included in this version of OpenVMS Alpha.

Table 4–5 Run-Time Libraries Not Included on OpenVMS Alpha Systems

DBGSSISHR	DEBUG item, replaced by SYS$SSISHR on OpenVMS Alpha
DNS$RTL	No DNS in OpenVMS Alpha
DNS$SHARE	No DNS in OpenVMS Alpha
VBLAS1RTL	No support for VAX vector programs
VMTHRTL	No support for VAX vector programs

Most run-time libraries that were available in OpenVMS VAX Version 5.5-2 are available in this version of OpenVMS Alpha. The OpenVMS VAX Version 5.5-2 libraries that are not available are either not being ported to OpenVMS Alpha or are planned for a later release of OpenVMS Alpha.

For example, the vector math libraries VBLAS1RTL and VMTHRTL are not available in OpenVMS Alpha because there is no support on OpenVMS Alpha for programs that use the VAX vector instructions. ♦

4.25 Run-Time Library (LIB$)

This section contains release notes pertaining to the Run-Time Library (LIB$).

4.25.1 Corrections

This section describes corrections to the Run-Time Library.

4.25.1.1 LIB$SHOW_ZONE and Large Lookaside Lists

V7.0

Previously, calling LIB$SHOW_ZONE when the lookaside list contained more than 10,000 entries caused LIB$SHOW_ZONE to improperly report a LIB$_BADZONE error.

This problem has been fixed.

4.26 Screen Management (SMG$) Facility

The following sections contain release notes pertaining to the Screen Management (SMG$) Facility.

4.26.1 Changes and Enhancements

This section describes changes and enhancements to the software.

4.26.1.1 VT500 Series Terminal Support

V6.2

The Screen Management (SMG$) Facility implements support for VT500 series terminals through the use of a third group of characteristics. The following specific changes were made to SMG$ to support VT500 series terminals.

New Boolean Capabilities

Two new Boolean capabilities have been added for use as TERMTABLE entries:

OpenVMS Name	Used by SMG	Description
ANSI_COLOR	N	If set, terminal conforms to ANSI color programming standards
DEC_CRT_5	N	If set, terminal conforms to DIGITAL VT500 family standards

Changes to SMG$ Routines

The following SMG$ routines have been changed as described below to support VT500 series terminals:

SMG$GET_KEYBOARD_ATTRIBUTES The following new symbolic name provides access to VT500 series keyboard information table values through the **keyboard-info-table** argument.

SMG$L_DEV_DEPEND3	Specific characteristics 3 (longword).

SMG$GET_PASTEBOARD_ATTRIBUTES The following new symbolic name provides access to VT500 series pasteboard attribute values through the **pasteboard-info-table** argument.

SMG$L_DEVDEPEND3	Specific characteristics 3 (longword).

SMG$SET_TERM_CHARACTERISTICS Three new optional arguments change terminal characteristics and retrieve the current terminal characteristics for VT500 series terminals. The three arguments are **on-characteristics3**, **off-characteristics3**, and **old-characteristics3**.

The new syntax is:

```
SMG$SET_TERM_CHARACTERISTICS pasteboard-id
     [,on-characteristics1] [,on-characteristics2]
     [,off-characteristics1] [,off-characteristics2]
     [,old-characteristics1] [,old-characteristics2]
     [,on-characteristics3] [,off-characteristics3]
     [,old-characteristics3]
```

The three new arguments are defined as follows:

on-characteristics3
mask_longword
longword (unsigned)
read only
by reference
Bit mask that specifies the terminal characteristics to be set from $TT3DEF. The **on-characteristics3** argument is the address of an unsigned longword that contains the bit mask.

off-characteristics3
mask_longword
longword (unsigned)
read only
by reference
Bit mask that specifies the terminal characteristics to be reset from $TT3DEF. The **off-characteristics3** argument is the address of an unsigned longword that contains the bit mask.

old-characteristics3
mask_longword
longword (unsigned)
read only
by reference
Retrieves the current terminal characteristics in the third group. The **old-characteristics3** argument is the address of an unsigned longword that contains the bit mask.

4.27 Symbiont/Job Controller Interface (SMB) Routines

This section contains release notes pertaining to the Symbiont/Job Controller Interface (SMB) routines.

4.27.1 Changes and Enhancements

The following note describes a change to one of the Symbiont/Job Controller Interface (SMB) routines.

4.27.1.1 SMB$READ_MESSAGE_ITEM Routine Has a New Symbol

V7.0 The SMB$READ_MESSAGE_ITEM routine of the Symbiont/Job Controller Interface (SMB) has a new symbol, SMBMSG$V_NO_INITIAL_FF, for the SMBMSG$K_PRINT_CONTROL message item code. This symbol is defined as follows:

Symbol	Description
SMBMSG$V_NO_INITIAL_FF	The symbiont suppresses the initial form feed if this bit is turned ON.

4.28 System Services

The following sections contain release notes pertaining to system services.

All system services are documented in the *OpenVMS System Services Reference Manual*.

4.28.1 Problems and Restrictions

This section describes problems and restrictions related to system services.

4.28.1.1 Linking SECURESHR Images to Run on Older Versions

V7.0 Some additional entry points have been added to the shareable image dispatch vector. Because of this change, any applications linked against Version 7.0 of SECURESHR will not run on a pre-Version 7.0 system. System services that use SECURESHR are the following:

$FORMAT_ACL
$PARSE_ACL
$FORMAT_AUDIT
$DELETE_INTRUSION
$SCAN_INTRUSION
$SHOW_INTRUSION
$ADD_PROXY
$DELETE_PROXY
$DISPLAY_PROXY
$VERIFY_PROXY

If your program uses any of these system services and you want to create a version that runs on systems prior to Version 7.0, you must link your program on a system running a version of OpenVMS prior to Version 7.0.

4.28.1.2 $SUSPND Behaves Incorrectly in a Cluster Environment

VAX V6.0
Alpha V1.5

When the $SUSPND system service is called and the target process is on a different cluster node than that of the process calling the $SUSPND service, the kernel mode suspend flag (bit 0) is ignored. As a result, any suspend is treated as a supervisor-mode suspend.

Digital expects to fix this problem in a future release.

4.29 System Time Operation

The following sections contain release notes pertaining to the system time operation.

4.29.1 Documentation Changes and Corrections

This section includes new documentation information.

4.29.1.1 *OpenVMS Programming Concepts Manual*

V6.1

The following information should be included in the "System Time Operations" chapter of the *OpenVMS Programming Concepts Manual*.

The Date/Time Manipulation option provides date/time spelling support for four new languages. Users or application programmers can select the desired language by defining the logical name SYS$LANGUAGES. The new languages and their equivalent names are as follows:

Language	Equivalent Name
Chinese (simplified character)	Hanzi
Chinese (traditional character)	Hanyu
Korean	Hangul
Thai	Thai

Defining Date/Time Spelling

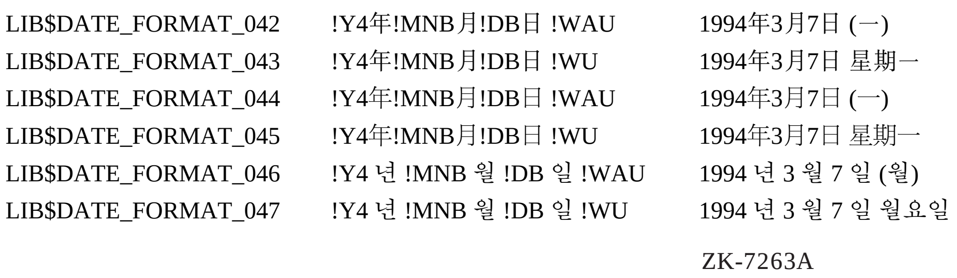
To define the spelling for Hanzi and Hanyu, define SYS$LANGUAGES as shown below, prior to invoking LIB$DT_STARTUP.COM:

```
$ DEFINE SYS$LANGUAGES HANZI, HANYU
$ @SYS$MANAGER:LIB$DT_STARTUP
```

Predefined Output Formats

Tables 4–6 and 4–7 list the new predefined date and time format logical names, their formats, and examples of the output generated using those formats.

Table 4–6 Predefined Output Date Formats



Logical Name	Format	Example
LIB$DATE_FORMAT_042	!Y4年!MNB月!DB日 !WAU	1994年3月7日 (一)
LIB$DATE_FORMAT_043	!Y4年!MNB月!DB日 !WU	1994年3月7日 星期一
LIB$DATE_FORMAT_044	!Y4年!MNB月!DB日 !WAU	1994年3月7日 (一)
LIB$DATE_FORMAT_045	!Y4年!MNB月!DB日 !WU	1994年3月7日 星期一
LIB$DATE_FORMAT_046	!Y4 년 !MNB 월 !DB 일 !WAU	1994 년 3 월 7 일 (월)
LIB$DATE_FORMAT_047	!Y4 년 !MNB 월 !DB 일 !WU	1994 년 3 월 7 일 월요일

ZK-7263A

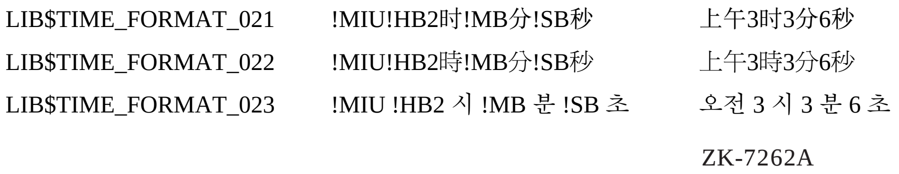
Note

LIB$DATE_FORMAT_042 and LIB$DATE_FORMAT_043 support the DEC Hanzi coded character set.

LIB$DATE_FORMAT_044 and LIB$DATE_FORMAT_045 support the DEC Hanyu coded character set.

LIB$DATE_FORMAT_046 and LIB$DATE_FORMAT_047 support the DEC Hangul coded character set.

Table 4–7 Predefined Output Time Formats



Logical Name	Format	Example
LIB$TIME_FORMAT_021	!MIU!HB2时!MB分!SB秒	上午3时3分6秒
LIB$TIME_FORMAT_022	!MIU!HB2時!MB分!SB秒	上午3時3分6秒
LIB$TIME_FORMAT_023	!MIU !HB2 시 !MB 분 !SB 초	오전 3 시 3 분 6 초

ZK-7262A

Note

LIB$TIME_FORMAT_021 supports the DEC Hanzi coded character set.

LIB$TIME_FORMAT_022 supports the DEC Hanyu coded character set.

LIB$TIME_FORMAT_023 supports the DEC Hangul coded character set.

Thus, to select a particular format for a date or time, or both, you can define the LIB$DT_FORMAT logical name using the following logicals:

- LIB$DATE_FORMAT_*nnn*, where *nnn* can range from 001 to 047
- LIB$TIME_FORMAT_*nnn*, where *nnn* can range from 001 to 023

4.30 VAX MACRO (VAX Only)

VAX The following sections contain release notes pertaining to VAX MACRO.

4.30.1 Documentation Changes and Corrections

This section describes corrections to the VAX MACRO documentation.

4.30.1.1 *VAX MACRO and Instruction Set Reference Manual*

V6.1 The following sections describe corrections to the *VAX MACRO and Instruction Set Reference Manual*.

4.30.1.1.1 .ASCIZ Macro Directive

V6.1 Chapter 6 of the *VAX MACRO and Instruction Set Reference Manual*, Version 6.0 describes the .ASCIZ macro directive. In the example that illustrates the .ASCIZ directive, the following line of code is incorrect:

```
.ASCIZ /A/<KEY>(FF\TEXT)/B/       ; 3 characters in string,
                              ;  4 bytes of data
```

Change this code to:

```
.ASCIZ /A/<FF>/B/        ; 3 characters in string
```

4.30.1.1.2 General Register Addressing

V6.1 Chapter 8 of the *VAX MACRO and Instruction Set Reference Manual*, Version 6.0 describes the general register addressing notations for the Macro assembler. In Table 8-5, the assembler notation for “Register deferred” (addressing mode 6) in Macro code is listed as:

Rn

Change to:

(Rn)

4.30.1.1.3 VAX Instruction CASE

V6.1 Chapter 9 of the *VAX MACRO and Instruction Set Reference Manual*, Version 6.0 describes the VAX instruction set. Where the description of the CASE instruction references symbol **disp1**[0] twice in the first note, the symbols erroneously use the digit 1 (one) instead of the letter “l”.

Change both occurrences of **disp1**[0] to:

displ[0] ♦

4.31 X/Open Transport Interface (XTI)

The notes in this section describe the X/Open Transport Interface (XTI).

4.31.1 Changes and Enhancements

V6.2 OpenVMS Version 6.2 supports the X/Open Transport Interface (XTI) programming interface. The implementation conforms with the XPG4 X/Open CAE XO/CAE/91/600 (ISBN 1 872630 29 4) X/Open Transport Interface (XTI) specification.

Supported Transports

OpenVMS Version 6.2 supports the DECnet for OpenVMS (Phase IV) and TCP/IP transports. See Section 4.31.2 for support restrictions.

The transport names used in the t_open routine are 'dnet' for DECnet for OpenVMS, and either 'ip/udp' or 'ip/tcp' for TCP/IP.

Other transports are available with other networking layered products. Consult individual layered product documentation for information about OpenVMS XTI support.

Architecture

XTI is supported by front end and back end code. Front end code provides access to the standard interface routines. Back end code provides the interface from the front end code to the selected networking transport.

The supporting image files are as follows:

XTI front end code	SYS$SHARE:XTI$XTILIB.EXE
TCP/IP XTI back end code	SYS$SHARE:XTI$IPSHR.EXE
DECnet for OpenVMS XTI back end code	SYS$SHARE:XTI$DNETSHR.EXE
XTI C programming include file	SYS$LIBRARY:XTI.H

Linking Requirements

After compiling an XTI program, no additional qualifiers are required for linking with XTI.

Documentation

Documentation about XTI is not included in the OpenVMS Version 6.2 release. You can order documentation directly from X/Open Company Limited. If you have access to the Internet, you can get more information about X/Open Company Limited (including their publications) by browsing the following URL:

http://www.xopen.co.uk/

You can also contact X/Open Company Limited at the following locations:

- USA: East Coast

  X/Open Company Limited
  3141 Fairview Park Drive
  Falls Church
  VA 22042-4501
  U.S.A.
  Tel: +1 (703) 876 0044
  Fax: +1 (703) 876 0050

- USA: West Coast

  X/Open Company Limited
  1010 El Camino Real
  Suite 380
  Menlo Park, CA 94025
  U.S.A.
  Tel: +1 (415) 323 7992
  Fax: +1 (415) 323 8204

- United Kingdom:

  X/Open Company Limited
  Apex Plaza
  Forbury Road
  Reading
  Berks RG1 1AX
  U.K.
  Tel: +44 1734 508311
  Fax: +44 1734 500110

- Japan:

  X/Open Company Limited
  Karufuru-Kanda Bldg, 9F
  1-2-1 Kanda Suda-cho
  Chiyoda-Ku
  Tokyo 101
  Japan
  Tel: +81 3 3251 8321
  Fax: +81 3 3251 8376

4.31.2 Problems and Restrictions

V6.2 The following restrictions apply to the OpenVMS Version 6.2 XTI:

- Nonblocking I/O is unsupported for DECnet for OpenVMS.

  DECnet for OpenVMS (Phase IV) does not fit the model for nonblocking I/O. Attempts to open or switch an XTI file descriptor to nonblocking I/O (O_NONBLOCK) will fail.

- Connectionless I/O is unsupported for DECnet for OpenVMS.

  DECnet for OpenVMS (Phase IV) does not fit the model of connectionless I/O. Therefore, only connection-oriented connections are supported.

- Disabled ASTs cause problems.

  The XTI back end code uses ASTs for asynchronous delivery of events from the transports. If ASTs are disabled (sys$setast(0)), the XTI back end code will not operate correctly until the ASTs are enabled again.

- XTI file descriptors are not compatible with C Run-Time Library file descriptors.

In addition, the ’t_info’ structure returned from the t_open function reports any additional implementation-specific restrictions for the given transport. (See the XTI documentation for information about the t_open command.)

A

Remedial Kits Included in OpenVMS Version 7.0

This appendix lists remedial kits that are included in OpenVMS Version 7.0. Appendix B describes other remedial kits that are *not* included in Version 7.0, but that will be available in a future release.

Digital updates existing kits and creates new kits as necessary. Contact your Digital support representative for the latest information about new remedial kits.

The following sections list the remedial kits included in Version 7.0 of the OpenVMS VAX and OpenVMS Alpha operating systems. If you used to install one of the kits listed here, you will not need to do so with Version 7.0.

Kit names are constructed from the following information in this order:

- Platform name: VAX or ALP (for Alpha)
- Facility name, abbreviated to 4 characters if necessary
- Number of the kit for this facility for this version
- Version number

For example, VAXQMAN01_062 is the first remedial kit created to correct the queue manager that shipped in Version 6.2 of OpenVMS VAX.

A.1 Remedial Kits Included in OpenVMS VAX Version 7.0

VAX

The following remedial kits are included in Version 7.0 of the OpenVMS VAX operating system:

VAXCLIU01_062
VAXF11C01_062
VAXF11X01_062
VAXF11X02_062
VAXINST01_062
VAXLAVC01_061
VAXLAVC04_U2055
VAXLOGI01_062
VAXMTAA03_062
VAXPRTS03_062
VAXQMAN01_062
VAXSALV01_062
VAXSHAD01_062
VAXSYS01_062
VAXSYS02_062
VAXSYS03_062
VAXSYS13_U2055
VAXSYS15_061

VAXVERI01_062 ♦

A.2 Remedial Kits Included in OpenVMS Alpha Version 7.0

Alpha

The following remedial kits are included in Version 7.0 of the OpenVMS Alpha operating system:

ALPAPB01_062
ALPBASR01_062
ALPCLIU01_062
ALPCPUC01_062
ALPCPU001_062
ALPF11C02_062
ALPF11X01_062
ALPINST01_062
ALPLAVC01_062
ALPLIBR2_062
ALPMAIL01_061
ALPMTAA01_062
ALPOPDR01_062
ALPOPDR05_061
ALPPRTS01_062
ALPQMAN01_062
ALPSALV01_062
ALPSCSI02_062
ALPSCSI03_062
ALPSHAD01_062
ALPSYS01_062
ALPSYS02_062
ALPSYS14_061
ALPSYSL01_062
ALPTRAC01_062
ALPTRAC02_061
ALPTTDR01_062
ALPVERI01_062
AXPDRIV01_062
AXPLAN01_062
AXPSCSI01_062
AXPTRAC01_061 ♦

B
Remedial Kits Not Included in OpenVMS Version 7.0

This appendix describes remedial kits that are *not* included in OpenVMS Version 7.0. Appendix A lists all remedial kits that *are* included in Version 7.0 of the OpenVMS operating system.

Kit names are constructed from the following information in this order:

- Platform name: VAX or ALP (for Alpha)
- Facility name, abbreviated to 4 characters if necessary
- Number of the kit for this facility for this version
- Version number

For example, ALPLAD02_062 is the second remedial kit created to ammend the LASTCP facility that shipped with Version 6.2 of OpenVMS Alpha.

Digital updates existing kits and creates new kits as necessary. Patch kits are available through the following distribution channels:

- **Customer Support Centers (CSC)**

  In the United States, call 1-800-354-9000.

  In other countries, contact your local CSC.

- **World Wide Web Sites**

  Look for OpenVMS kits at this URL:

  `http://www.service.digital.com/html/patch_main.html`

- **File Transfer Protocol (FTP)**

  OpenVMS VAX and Alpha kits are available from:

  `ftp.service.digital.com`

B.1 Remedial Kits Not Included in OpenVMS VAX Version 7.0

VAX

The following sections describe remedial kits that are *not* included in Version 7.0 of the OpenVMS VAX operating system. These kits will likely be included in the next release of the operating system.

B.1.1 Record Management Services (RMS) Kit (VAXRMS02_061)

This remedial kit corrects the following problems:

- Heavy, concurrent use of the INSTALL command and the DCL lexical function F$FILE_ATTRIBUTE can cause locking conflicts while attempting to access the Known File Entry (KFE) list. This conflict can result in a bad KFE pointer being sent to RMS, which causes an ACCVIO error accompanied by the following RMS error:

  ```
  SSRVEXCEPT, unexpected system service exception
  ```

- Incomplete security success audit information is sometimes generated upon image activation of installed /OPEN images.
- The following fatal LIB error is reported while converting an input VFC-format file to an output fixed-format file using the /PAD qualifier:

  ```
  BADBLOADR, bad block address
  ```

  This error was introduced to the Convert utility in OpenVMS Version 6.1.

B.1.2 System Services Kit (VAXSYS04_062)

This kit corrects an error in the $GETDVI system service that causes the DCL command ALLOCATE/GENERIC to erroneously return the following SYSTEM message whether or not the device is found:

```
NOSUCHDEV,  no such device available
```

After the remedial kit is applied, the correct message is returned under the appropriate conditions. ♦

B.2 Remedial Kits Not Included in OpenVMS Alpha Version 7.0

Alpha

The following sections describe remedial kits that are *not* included in Version 7.0 of the OpenVMS Alpha operating system. These kits will likely be included in the next release of the operating system.

B.2.1 LAST Control Program (LASTCP) Kit (ALPLAD02_062)

This kit resolves disk corruption problems that occur when you use the LADDRIVER component of the InfoServer client with PATHWORKS components such as the PCDISK utility, and then connect to the same service by executing the USE command from a PC.

B.2.2 SCSI Driver Kit (ALPSCSI03_061)

This kit corrects a failure that occurs in PKCDRIVER when any SCSI target device takes longer than two seconds to complete a command. The command is aborted and the system error log is updated with a timeout error (and possibly an unexpected interrupt error). The errors imply a target failure when there really is no problem.

The remedial kit changes PKCDRIVER to use the DMA Timeout field of the SCSI Class Driver Request Packet (SCDRP) instead of the former fixed two-second timeout value. This fix allows SCSI class drivers to specify whatever timeout value is required to complete the command and avoid the error. ♦

Index

A

Account removal problem
  with POLYCENTER Software Installation utility, 4–54
Ada Boolean expressions, 4–9
Ada enumeration variables, 4–9
AGEN$PARAMS.REPORT
  managing, 3–2
ALPHA$LIBRARY:STARLET.MLB file, 4–50
AlphaServer 2100 systems
  SCSI controllers, 1–10
ALPHAVMSSYS.PAR file
  VMScluster system startup problem, 3–45
ANALYZE/AUDIT command
  /IGNORE qualifier
    SUBTYPE criterion, 3–1
  /SELECT qualifier
    SUBTYPE criterion, 3–1
Archive file
  creating for each node in a cluster, 3–33
Arrays
  COBOL INITIALIZE statement, 4–2, 4–10
  depositing values, 4–2
  EXAMINE command, 4–3
  variable used as index, 4–9
.ASCIZ directive
  Macro assembler, 4–64
Atomicity alignment considerations
  correction to documentation example, 4–51
Audit Analysis utility
  documentation corrections, 3–1
Authorize utility
  WSEXTENT qualifier, 3–2
AUTOGEN
  changes and enhancements
    calculating page file size on Alpha, 3–3
    computing paging file size on VAX, 3–4
    computing VIRTUALPAGECNT, 3–4
    computing WSMAX, 3–4
    managing AGEN$PARAMS.REPORT, 3–3
    NPAGEDYN, 3–3
    NPAGEVIR, 3–3
    PQL_DPGFLQUOTA, 3–3
    PQL_MPGFLQUOTA, 3–3
    VIRTUALPAGECNT, 3–4
  problems and restrictions
AUTOGEN
  problems and restrictions (cont'd)
    full memory dumps, 3–5
AXPEMUL01_070 kit, 1–6

B

Backup utility
  problems and restrictions
    standalone BACKUP, 3–7
    VMS$COMMON.DIR file, 3–8
Backup utility (BACKUP)
  problems and restrictions
    CD–ROM menu system, 3–6
    image backup from RF73, 3–7
    incremental backups, 3–5
    incremental backups using PATHWORKS, 3–5
    warnings on blocks incorrectly marked, 3–6
  relative file versions, 3–7
  saving and restoring alias directories, 3–8
  system disk backup, 3–6
Batch and print queues
  problems and restrictions
    PRINT/DELETE command, 3–9
    SUBMIT/DELETE command, 3–9
    terminating batch jobs, 4–1
  process rights identifier limit, 3–9
Battery-Backed Watch (BBW)
  setting time using, 3–40
Before-image journaling
  restriction, 3–33
Booting
  TURBOchannel devices, 3–35
Booting the system
  informational messages, 1–9
Branch
  target of indeterminate, 4–52

C

C structure member names
  passing restriction, 4–40
CANCEL SELECTIVE function
  improved use with LTDRIVER, 4–48

- CASE instruction, 4–64
- CD–ROM menu system, 3–6
- CHANNELCNT system parameter, 4–14
- CHKABSADR message
  - suppressing during MACRO-32 compilation, 4–53
- CI lines unsupported for DECnet communication, 3–18
- Circumflex (^)
  - use restriction, 4–50
- CLI (command line interpreter), 4–5
- CLUSTER_CONFIG.COM
  - TMSCP_SERVE_ALL parameter, 3–44
- CLUSTER_CONFIG_LAN.COM
  - TMSCP_SERVE_ALL parameter, 3–44
- CMA interface, 4–26
- COBOL INITIALIZE statement and large tables (arrays), 4–2, 4–10
- COBOL module name, 4–10
- COBOL scaled computational data, 4–10
- COBOL variable name, 4–10
- COM$DRVDEALMEM routine
  - synchronization, 4–42
- Command line interpreter, 4–5
- Concurrency
  - improved in VMScluster systems, 3–44
- CRB$L_INTD field
  - interrupt vector, 4–42

D

- DATA keyword
  - workaround for linker overlay restriction, 4–47
- Data loss
  - RF73 and RF*nn* controller memory errors, 1–12
- Date/time formats, 4–62
- DCL commands
  - changes and enhancements
    - ANALYZE/ERROR_LOG/SUMMARY command, 3–10
    - SHOW NETWORK command, 3–10
  - errors with lexical functions, 3–8
  - problems and restrictions
    - SET PROCESS/NOAUTO_UNSHELVE, 2–1
    - SET PROCESS/SUSPEND=KERNEL/ID, 2–1
- DCL Help
  - affected during layered product installation, 1–4, 1–5
- DDCMP lines, 3–18
- DEBUG option, 4–53
- /DEBUG qualifier, 4–53
- DEBUG.EXE name, 4–4
- Debugger
  - See OpenVMS Debugger, Delta/XDelta debugger, Kept Debugger
- DEC 2000 Model 300, 4–46
- DEC 3000 workstations
  - PXG graphics board not supported, 1–10
- DEC Ada Run-Time Library
  - unexpected storage errors, 4–12
- DEC C RTL
  - changes and enhancements, 4–13
    - BSD curses, 4–18
    - date and time support, 4–16
    - floating-point values, 4–18
    - fscanf( ) function changes, 4–17
    - ICONV utility enhancements, 4–15
    - internationalization support, 4–18
    - LOCALE utility, 4–18
    - mapping record carriage control data, 4–13
    - math functions, 4–19
    - message specifications relaxed, 4–17
    - open file limit increased, 4–14
    - open routines and RMS options, 4–19
    - open socket support increased, 4–14
    - setvbuf function, 4–19
    - sleep function, 4–19
    - stat function, 4–19
    - summary of new functions, 4–14
    - wildcards in file specifications, 4–16
    - writing to pipes, 4–20
  - corrections
    - file protection status mapped to EACCES, 4–21
    - invalid RMS options, 4–21
    - LOCALEDEF utility, 4–21
    - realloc( ) function, 4–22
    - relative pathnames translation, 4–21
    - socket information, 4–21
    - SYS_ERRLIST, 4–22
    - SYS_NERR, 4–22
  - problems and restrictions, 4–20
    - DECwindows Motif compatibility problem, 4–20
    - linking with /NOSYSSHR, 4–20
    - socket behavior, 4–20
    - using /NOSYSSHR to link DEC C applications, 4–20
- DEC C++
  - problems and restrictions, 4–12
- DEC C++ Class Library
  - corrections, 4–13
  - threadsafe support, 4–12
- DEC Fortran
  - See Fortran
- DEC Notes
  - problem with DECwindows interface, 2–3
- DEC Text Processing Utility (DECTPU)
  - DECwindows Motif interface
    - Help built-in disabled, 2–3
    - small display monitors, 4–31

- DECamds
  - changes and enhancements
    - support removed for old VAX versions, 3–13
    - VERUNSUP error message, 3–13
  - corrections
    - collecting image names, 3–14
    - identifying disks in a VMScluster, 3–14
    - saving resource hash table dense/sparse thresholds, 3–14
  - problems and restrictions
    - DECwindows Motif support, 3–14
    - kernel threads unsupported on Alpha, 3–14
- DECdtm services
  - in a DECnet/OSI network, 3–14
  - problems and restrictions
    - SCSNODE values, 3–15
  - transaction group
    - example, 3–15
- DECevent, 3–17
  - documentation location, 3–17
  - problems and restrictions
    - device support, 3–17
    - using logical file names, 3–17
- DECforms
  - support on OpenVMS Alpha Version 7.0, 2–2
- DECnet
  - DECnet/OSI supersedes DECnet Phase IV, 2–2
  - product requirements, 1–4
  - specifying transport, 2–5
  - using on Token Ring devices, 1–9
- DECnet for OpenVMS Alpha
  - restrictions, 3–17
- DECnet for OpenVMS VAX (Phase IV)
  - problems and restrictions, 3–18
- DECnet/OSI for OpenVMS Alpha, 3–19
- DECnet/OSI for OpenVMS VAX, 3–19
- DECnet/OSI network
  - DECdtm services in, 3–14
- DECpresent
  - installation dependencies with OpenVMS VAX, 1–6
- DECram
  - Version 2.2 unsupported on Alpha, 1–8
- DECserver 700 terminal controller
  - increased baud rate support on LAT, 1–14
- DECthreads
  - changes and enhancements
    - application coding errors, 4–26
    - CMA interface, 4–26
    - DECthreads debugger, 4–28
    - exit handling, 4–27
    - $EXIT system service, 4–27
    - $HIBER, 4–28
    - latent bugs, 4–26, 4–27
    - multiprocessor support, 4–27
    - POSIX 1003.1c interface, 4–25
- DECthreads
  - changes and enhancements (cont'd)
    - POSIX 1003.4a Draft 4 interface, 4–26
    - system services blocked, 4–28
    - Thread Independent Services (TIS) interface, 4–26
    - timed condition waits, 4–26
  - online help, 4–31
  - problems and restrictions
    - 64-bit parameters, 4–29
    - condition variable attributes, 4–29
    - DECthreads debugger metering function, 4–29
    - dynamic image activation, 4–30
    - errno value, 4–29
    - exception handling, 4–29
    - exit handler routine, 4–30
    - language support, 4–28
    - not supported on POSIX for OpenVMS, 4–29
    - OpenVMS Debugger, 4–29
    - release compatibility, 4–28
    - thread terminations, 4–30
    - using Debugger SET TASK/ACTIVE command, 4–29
- DECTPU
  - See DEC Text Processing Utility
- DECTPU SET built-ins
  - WIDGET_CONTEXT_HELP keyword, 2–3
- DECwindows Motif, 4–5
  - creating WSA pseudo workstation devices, 3–23
  - DEC Notes problem, 2–3
  - height and width restrictions, 3–22
  - LinkWorks error message, 2–3
  - patch kit on CD–ROM, 2–3
  - problems and restrictions
    - console broadcasts, 3–21
    - system files, 3–21
    - Version 1.2 compatibility problem, 3–20
    - virtual data error message, 3–21
  - TCP/IP support for NCSA Mosaic browser, 2–3
  - Version 1.1 for OpenVMS not supported, 2–3
  - Version 1.2 restrictions
    - World Wide Support, 3–22
- DEFQA controller
  - NETCONFIG.COM support, 3–18, 3–19
- DEFTA controller, 3–18
  - NETCONFIG.COM support, 3–19
- Delta/XDelta debugger
  - base register default offset, 4–32
  - floating point register support, 4–32
  - image activator, 4–31
- DEPOSIT command, 4–10
- Deposits to shared data, 4–11
- Device driver coding
  - MASSBUS adapters, 4–41

- Device support
  - Step 2 driver interface, 4–32
  - user-written drivers, 4–32
  - writing drivers in C, 4–32
  - writing drivers in Macro-32, 4–32
- Digital 2100 Server Model A500/600MP, 4–46
- Digital DCE for OpenVMS
  - support restrictions, 3–23
- Digital Product Performance Program, 3–26
- /DISABLE qualifier, 4–53
- Disks
  - RF73 and RF*nn*
    - controller memory errors, 1–12
- DISMOUNT/CLUSTER command
  - volume shadowing restriction, 3–48
- Documentation changes and corrections
  - *A Comparison of System Management on OpenVMS AXP and OpenVMS VAX* retired, 3–10
  - *Migrating an Application from OpenVMS VAX to OpenVMS Alpha* (new), 4–54
  - *Migrating an Environment from OpenVMS VAX to OpenVMS Alpha* (new), 3–10
  - *Migrating to an OpenVMS AXP System: Planning for Migration* retired, 4–54
  - *Migrating to an OpenVMS AXP System: Recompiling and Relinking Applications* retired, 4–54
  - *OpenVMS Compatibility Between VAX and Alpha* retired, 3–10
  - *OpenVMS DCL Dictionary*, 3–11
  - *OpenVMS Delta/XDelta Debugger Manual*, 4–32
  - *OpenVMS I/O User's Reference Manual*, 4–46
  - *OpenVMS Linker Utility Manual*, 4–48
  - *OpenVMS Programming Concepts Manual*, 4–62
  - *OpenVMS System Management Utilities Reference Manual: A–L*, 3–1
  - *OpenVMS System Manager's Manual: Essentials*, 3–29, 3–40
  - *OpenVMS System Manager's Manual: Tuning, Monitoring, and Complex Systems*, 3–33
  - *OpenVMS VAX Device Support Reference Manual*, 4–42
  - *POLYCENTER Software Installation Utility Developer's Guide*, 4–57
  - *Porting VAX MACRO Code from OpenVMS VAX to OpenVMS Alpha*, 4–50
  - *VAX MACRO and Instruction Set Reference Manual*, 4–64
- DR_UNIT_BASE system parameter, 1–9
- DSSI disk devices
  - microcode revision levels, 1–12
- DUMP command qualifiers
  - /DESCRIPTOR, 3–11
  - /MEDIA_FORMAT, 3–11
  - /PATH_TABLE, 3–11
- Dump files
  - See System dump files
- Dump Off System Disk (DOSD), 3–43

E

- EDIT command, 4–5
- EFN$C_ENF event flag, 4–43
- /ENABLE qualifier, 4–53
- Entry points
  - when to declare, 4–52
- $EQULST macro
  - symbol value, 4–43
- Error messages
  - Sort/Merge utility (high-performance), 2–4, 4–44
- Error Reporting Formatter (ERF), 3–17
- Ethernet controllers
  - problems with multiple controllers, 1–10
- EVALUATE command and truncation, 4–10
- Event flags
  - EFN 128, 4–43
- EXAMINE command and large arrays, 4–3
- EXAMINE LABEL command, 4–7
- Exit handling, 4–27
- $EXIT system service, 4–27
- Exiting a window, 4–5
- Extended DDT bit
  - problem corrected, 4–48

F

- Fault management tools, 3–17
- FDDI clusters
  - restriction removed, 3–44
- FDDI controllers
  - problems with multiple controllers, 1–10
- FDDI to Q–bus controller
  - NETCONFIG.COM support, 3–18, 3–19
- FDDI to TURBOchannel controller
  - NETCONFIG.COM support, 3–19
- FDDI to TURBOchannel controller (DEFTA), 3–18
- File extension, default
  - Sort/Merge utility (high-performance), 2–4, 4–44
- /FLAGGING=INSTRUCTIONS qualifier, 4–53
- Floating point register support
  - Delta/XDelta debugger, 4–32
- Floating-point return values
  - requirement for "jacket", 4–49
- Fortran
  - Mathematics RTL interoperability restrictions, 4–53
- Fortran complex numbers, 4–10

G

General register addressing
  assembler, 4–64
Global pages
  required by debugger, 3–13
Global symbols table, 4–41
GO command, 4–11
Granularity preservation
  lost with INSV code, 4–50

H

Heap Analyzer
  and memory, 4–11
  and /NODEBUG, 4–3
Help Message utility (MSGHLP)
  restriction on user-supplied additions, 3–24
High-performance Sort/Merge utility, 2–4

I

I/O write operations
  failure, 4–45
ICONV utility enhancements, 4–15
Image Activator
  automatic activation with debugger, 4–31
Image registration
  and shareable images, 3–24
Incomplete %DEBUG-E-LINEINFO message,
  4–10
Incomplete information displayed, 4–11
InfoServer client software
  failure to start, 1–4
InfoServer software
  updating from the Software Products Library,
    1–3
Initialization
  problem with recovery of root directory, 3–24
Inlined routines, 4–8
Install utility (INSTALL)
  installing images change, 3–25
Installation and upgrade information
  Alpha only, 1–6
  VAX only, 1–5
Installation and upgrade restrictions
  DECpresent dependencies, 1–6
  layered product installations and DCL Help,
    1–5
  TCP/IP requirements, 1–4
  VAXstation and MicroVAX
    system hangs, 1–6
Instruction Emulator
  installing, 1–6
INSV instruction
  impact on granularity, 4–50
Internal lock manager access code
  required modification, 4–48
Internet
  specifying transport, 2–5
ISO 9660 standard
  dumping ISO 9660 volumes, 3–11
  using /CLUSTER qualifier in Mount utility,
    3–29
  using SVDs to mount volumes, 3–29

K

Kept Debugger, 4–7

L

LAT baud rate support increased, 1–14
Layered products
  installations with POLYCENTER Software
    Installation Utility, 1–4
  installations with VMSINSTAL, 1–5
  Software Public Rollout Reports, 2–5
  versions supported for OpenVMS Version 7.0,
    2–5
LCKMGR
  spin lock for standalone OpenVMS Alpha
    systems, 4–48
LIB$FIND_IMAGE_SYMBOL routine
  potential DECthreads problem, 4–30
LIB$SHOW_ZONE
  large lookaside lists, 4–59
Librarian utility (LIBRARIAN)
  error reporting problem and workaround, 4–46
  operation failure, 4–46
Library text modules, 4–11
Linker utility, 4–47, 4–48
  linking with MTHRTL, 4–53
  new error message, 4–47
  workaround for restricted use of global symbols,
    4–47
Linking device driver images, 4–41
Lock manager
  synchronization changes, 4–48
Logging
  turning off, 3–2
Logical names
  MAIL$INTERNET_MODE, 2–5
  MAIL$INTERNET_TRANSPORT, 2–5
Long text strings
  improper display, 4–9
LTDRIVER restriction, 4–48

M

- Macro directive
  - .ASCIZ, 4–64
- MACRO–32 compiler
  - atomicity alignment considerations, 4–51
  - BUG*x* instruction not supported, 4–52
  - DEBUG option restriction removed, 4–53
  - documentation changes and corrections, 4–50
  - entry point declarations, 4–52
  - /FLAGGING=INSTRUCTIONS function, 4–53
  - floating-point instruction restriction, 4–51
  - maintaining common sources, 4–50
  - packed-decimal instruction restriction, 4–51
  - problems and restrictions
    - .ASCID directive, 4–49
    - floating-point and packed-decimal instruction arguments, 4–50
    - floating-point return values in R0, 4–49
    - .GLOBAL_LABEL directive, 4–49
    - INSV instructions, 4–50
    - .JSB_ENTRY directive, 4–50
    - multiple object files compilation, 4–50
    - STARLET.MLB, 4–49
  - TRACEBACK option restriction removed, 4–53
- MACRO32.EXE
  - required by standalone BACKUP, 1–6
- Mail utility
  - changes and enhancements
    - SET FORWARD command, 2–6
    - transports, 2–5
    - use of privileges, 3–25
  - problems and restrictions
    - replying to an address containing nested quotation marks, 2–6
- Mail utility (MAIL)
  - problems and restrictions
    - replying to an address containing nested quotation marks, 3–26
- MAIL$INTERNET_MODE logical, 2–5
- MAIL$INTERNET_TRANSPORT logical, 2–5
- MAIL$SYSTEM_FLAGS logical, 2–6, 3–26
- MAIL.EXE
  - changes and enhancements
    - use of privileges, 3–25
- MASSBUS adapters
  - I/O registers, 4–41
- Microcode revision levels
  - commands for updating, 1–13
  - on DSSI disk devices, 1–12
- MicroVAX installation system hangs, 1–6
- Migration documentation
  - new and retired, 3–10
- MMOV$MSBDRIVER, 4–33
- MODPARAMS.DAT
  - logging statements, 3–2
- Monitor utility (MONITOR)
  - changes and enhancements
    - TCP/IP support, 3–28
  - display change on VAX, 3–28
  - monitoring remote disks
    - new limit, 3–28
- Monitoring Performance History (MPH), 3–26
- MOP (maintenance operation protocol)
  - incompatible with kernel threads, 3–19
- MOUNT command, 3–28
- Mount utility (MOUNT), 3–28
  - correction to noncompliant ISO 9660 CD–ROMs, 3–29
- MOUNT/FOREIGN command
  - limitation, 3–29
- MPH
  - See Monitoring Performance History
- MSGHLP
  - See Help Message utility
- MTH$ RTL
  - executable image restrictions, 4–53
- Multiprocess applications
  - inaccurate displays during debugging, 4–11
- Multiprocessor support for DECthreads, 4–27

N

- /NATIVE_ONLY qualifier, 4–48
- NCSA Mosaic browser
  - TCP/IP support options, 2–3
- NETCONFIG.COM
  - support for DEFQA controller, 3–18, 3–19
  - support for DEFTA controller, 3–19
- /NODEBUG qualifier and Heap Analyzer, 4–3
- Nonpaged pool
  - problems and restrictions
    - prepopulation, 3–30
- NPAGEDYN, 3–3
- NPAGEVIR, 3–3
- Null frame procedure descriptor
  - potential problem to debugger, 4–50
- Null frame procedures, 4–11

O

- OPCDEC condition, 4–11
- OPCOM
  - problems and restrictions
    - OPERATOR.LOG file, 3–31
- OpenVMS Alpha System-Code Debugger
  - problems, 4–40
- OpenVMS Alpha systems
  - booting messages, 1–9
  - compatibility with VAX, 3–9
  - RAID device-naming problems, 1–9
  - removing the operating system, 1–7

- OpenVMS Debugger
  - changes and enhancements
    - arrays, depositing values, 4–2
    - customization, 3–12
    - DECwindows Motif software requirement, 3–12
    - displaying a debug session from a PC, 3–12
    - internationalization, 3–12
    - resource file, 3–12
    - VMSDEBUG.DAT file, 3–12
  - corrections
    - Ada Boolean expressions, 4–9
    - Ada enumeration variables, 4–9
    - arrays, 4–9
    - arrays, large, and COBOL INITIALIZE statement, 4–10
    - COBOL INITIALIZE statement and arrays, 4–10
    - COBOL INITIALIZE statement and tables, 4–10
    - COBOL module name, 4–10
    - COBOL scaled computational data, 4–10
    - COBOL variable name, 4–10
    - DEPOSIT command, 4–10
    - deposits to shared data, 4–11
    - EVALUATE command and truncation, 4–10
    - Fortran complex numbers, 4–10
    - GO command, 4–11
    - Heap Analyzer and memory, 4–11
    - incomplete %DEBUG-E-LINEINFO message, 4–10
    - incomplete information displayed, 4–11
    - incorrect source code display, 4–11
    - incorrect text string display, 4–9
    - library text modules, 4–11
    - null frame procedures, 4–11
    - OPCDEC condition, 4–11
    - SEARCH command, 4–11
    - SET WATCH command, 4–11
    - STEP command, 4–11
    - variable used as array index, 4–9
    - windows interface customizations, 4–10
  - problems and restrictions
    - arrays, large, and EXAMINE command, 4–3
    - CLI, 4–5
    - COBOL INITIALIZE statement and large tables (arrays), 4–2
    - complex variables, 4–8
    - DEBUG.EXE name, 4–4
    - DECwindows Motif, 4–5
    - DECwindows Motif license requirements, 4–6
    - EDIT command, 4–5
    - errno value in threaded applications, 4–29
    - EXAMINE command and large arrays, 4–3

- OpenVMS Debugger
  - problems and restrictions (cont'd)
    - EXAMINE LABEL command, 4–7
    - exiting a window, 4–5
    - Heap Analyzer and /NODEBUG, 4–3
    - inlined routines, 4–8
    - Kept Debugger, 4–7
    - /NODEBUG qualifier and Heap Analyzer, 4–3
    - SHOW CALLS command, 4–5
    - static watchpoints, 4–4
    - STEP/INTO command, 4–8
    - STEP/OVER command, 4–8
    - system resource requirements, 3–13
    - system services, 4–4
    - user quotas, 3–13
    - using multiple kernel threads, 4–29
    - vector support, 4–9
    - watchpoint support, 4–5
    - watchpoints, static, 4–4
  - system management considerations, 3–12
- *OpenVMS Delta/XDelta Debugger Manual*
  - documentation corrections, 4–32
- *OpenVMS I/O User's Reference Manual*, documentation corrections, 4–46
- *OpenVMS Linker Utility Manual*, documentation corrections, 4–48
- *OpenVMS Programming Concepts Manual*, documentation corrections, 4–62
- *OpenVMS System Management Utilities Reference Manual: A–L*
  - documentation corrections, 3–1
- *OpenVMS System Manager's Manual: Essentials*
  - documentation corrections, 3–40
- *OpenVMS System Manager's Manual: Tuning, Monitoring, and Complex Systems*
  - documentation corrections, 3–33
- OpenVMS VAX systems
  - compatibility with Alpha, 3–9
  - standalone BACKUP requirement, 1–6
  - version requirements for upgrades, 1–5
- OpenVMS Version 6.2–*n* features, 1–2
- OpenVMS Version 7.0
  - features, 1–1
  - mixed-version clusters, 1–2
- Operating system
  - removing, 1–7

P

- Package operation constraints
  - with POLYCENTER Software Installation utility, 4–56
- Page file size
  - calculation changes, 3–3
- Paging file size computation, 3–4

- Parallel Processing Library (PPL) retirement, 4–58
- Per-disk licensing
  - volume shadowing, 3–48
- PGFLQUOTA
  - problems, 4–46
- POLYCENTER Software Installation utility
  - account removal problem, 4–54
  - alternate file placement restriction, 4–55
  - contention problem with DCL Help, 1–4
  - disk space reporting, 3–32
  - documentation corrections, 4–57
  - error message shortcoming, 4–55
  - error reporting, 4–55
  - execute statement, 4–56
  - file generations, 4–55
  - file statement documentation, 4–57
  - information statement, 4–56
  - package operation constraints, 4–56
  - partial kits, 4–56
  - PRODUCT command problems, 3–31
  - PRODUCT command restriction, 3–31
  - PRODUCT PACKAGE command, 4–57
  - removing products, 3–32
  - restriction in deleting directories, 3–31
  - rights identifier problem, 4–57
  - scope statement documentation, 4–57
  - scope support, 4–56
  - “uses” clause restriction, 4–57
- *POLYCENTER Software Installation Utility Developer’s Guide*
  - corrections, 4–57
- POLYCENTER Striping
  - replacement, 3–32
- Port driver $QIO
  - restriction, 4–48
- *Porting VAX MACRO Code from OpenVMS VAX to OpenVMS Alpha*
  - documentation changes, 4–50
- POSIX for OpenVMS
  - 1003.4a Draft 4 interface, 4–26
  - DECthreads interface, 4–25
  - not supported with DECthreads, 4–29
  - ordering Version 3.0, 3–32
  - Version 2.0 not supported, 3–32
- PPL (Parallel Processing Library) retirement, 4–58
- PQL_DPGFLQUOTA, 3–3
- PQL_MPGFLQUOTA, 3–3
- PQL_MWSEXTENT parameter settings, 3–2
- PRINT/DELETE command, 3–9
- Printing files, requirements, 3–9
- Privileges
  - with MAIL.EXE, 3–25
- Procedure descriptor, 4–52
- PRODUCT command restrictions, 3–31
- PRODUCT REMOVE command
  - files not removed with, 1–8
  - removing the operating system, 1–7
- PSECT keyword
  - component of linker overlay restriction, 4–47
- PXG graphics board not supported, 1–10

Q

- Quotas
  - required by debugger, 3–13

R

- RAID devices
  - naming problem, 1–9
- Record Carriage Control Data mapping, 4–13
- Recovery unit journaling
  - restriction, 3–33
- Register deferred
  - notation, 4–64
- Remedial kits
  - included in OpenVMS Alpha Version 7.0, A–2
  - included in OpenVMS VAX Version 7.0, A–1
  - not included in OpenVMS Alpha Version 7.0, B–2
  - not included in OpenVMS VAX Version 7.0, B–1
- Restrictions
  - debugger
    - system resource requirements, 3–13
    - user quotas, 3–13
  - KFESA/KFESB, 3–45
  - on passing C structure member names, 4–40
- RF73 disk
  - image backups of, 3–7
- RF*nn* disks
  - controller memory errors, 1–12
- Rights identifier problem
  - with POLYCENTER Software Installation utility, 4–57
- RMS Journaling
  - restriction, 3–33
- Root directory
  - incorrectly initialized, 3–24
- RRD45 CD–ROM testing, 3–42
- Run-Time Library (LIB$)
  - LIB$SHOW_ZONE, 4–59
- Run-time library (RTL) routines
  - not included in Alpha, 4–59
  - PPL retirement, 4–58
- RZ57 disk drives
  - volume shadowing support, 3–48

S

Screen Management (SMG$) Facility
  VT500 series terminal support, 4–59
SCSI controllers
  restrictions on AlphaServer 2100 systems, 1–10
SCSI disks
  volume shadowing limitation, 3–46
SCSI-2 port interface (SPI), 4–36
SCSI-2 status
  getting characteristics, 4–43
SDA dump file, 3–43
SEARCH command, 4–11
SECURESHR images, 4–61
Security auditing
  archive file for each node in a cluster, 3–33
  documentation corrections, 3–33
SET DISPLAY command
  creating WSA pseudo workstation devices, 3–23
SET FORWARD command, 2–6
SET PROCESS/NOAUTO_UNSHELVE command, 2–1
SET PROCESS/SUSPEND=KERNEL/ID= command, 2–1
SET WATCH command, 4–11
SHADZEROMBR error, 3–47
Shareable images and image registration, 3–24
Shared linkage sections
  restriction, 3–34
SHOW CALLS command, 4–5
SHOW DEVICE console command
  StorageWorks RAID Array 110 Subsystem, 3–35
SHOW NETWORK command
  new behavior, 3–10
SMB$READ_MESSAGE_ITEM routine, 4–61
SMBMSG$K_PRINT_CONTROL message item code, 4–61
SMBMSG$V_NO_INITIAL_FF symbol, 4–61
SMTP
  specifying transport, 2–5
Software Product Descriptions (SPDs), 1–3
Software Products Library
  using to update InfoServer software, 1–3
Software Public Rollout Reports, 2–5
Sort operations, concurrent
  Sort/Merge utility (high-performance), 2–4, 4–44
SORT32
  changes and enhancements
    work files, 3–34
Sort/Merge utility (high-performance)
  problems
    concurrent sort operations, 2–4, 4–44
    default file extension, 2–4, 4–44
Sort/Merge utility (high-performance)
  problems (cont’d)
    error messages, 2–4, 4–44
    merging stream files, 2–4
    merging stream files limitation, 4–44
Sort/Merge utility (SORT/MERGE)
  See Sort/Merge utility (high-performance)
Sound card problem, 4–37
Source code
  incorrect display, 4–11
  maintaining common VAX MACRO, 4–50
SPDs
  See Software Product Descriptions
SPI
  See SCSI-2 port interface
SPI$CONNECT macro
  R3 return with SPDT$M_CMDQ, 4–42
  R4 input, 4–42
SPI$GET_CONNECTION_CHAR macro
  characteristics buffer, 4–43
Standalone BACKUP, 3–7
  reliance on MACRO32.EXE, 1–6
STARLET.MLB, 4–49
  defining compiler directives, 4–50
Static watchpoints, 4–4
STEP command, 4–11
STEP/INTO command, 4–8
STEP/OVER command, 4–8
StorageWorks RAID Array 110 Subsystem
  SHOW DEVICE console command, 3–35
StorageWorks RAID Array 210 Subsystem
  naming problem, 1–9
StorageWorks RAID Array 230 Subsystem
  naming problem, 1–9
StorageWorks RAID Software for OpenVMS
  replacing POLYCENTER striping, 3–32
SUBMIT/DELETE command, 3–9
$SUSPND system service, 2–1
  cluster problem, 4–62
SVD (Supplementary Volume Descriptor)
  using to mount ISO 9660 volumes, 3–29
Symbiont/Job Controller Interface (SMB) routines, 4–61
SYS$LIBRARY:STARLET.MLB
  adding compiler directive definitions, 4–50
SYS$MSBDRIVER
  problem, 4–36
  removed from OpenVMS Distribution, 4–33
SYSGEN parameters
  MULTITHREAD, 3–19, 4–27
  PQL_MWSEXTENT, 3–2
SYSMAN
  See System Management utility
System Dump Analyzer (SDA) utility
  possible system failure, 3–43

System dump files
  Dump Off System Disk (DOSD), 3–43
  limitation, 3–5
  sizing, 3–4
System Management utility (SYSMAN)
  ALF command changes, 3–35
  changes and enhancements
    clusterwide DISKQUOTA commands, 3–38
  DISKQUOTA commands, 3–38
  problems and restrictions, 3–38
  rights identifier changes, 3–37
System monitoring
  See Monitoring Performance History
System parameters
  changes and corrections, 3–39
  TMSCP_SERVE_ALL, 3–44
  XQPCTL2, 3–44
System services
  blocking calls, 4–28
  linking SECURESHR images, 4–61
  supported by SSI, 4–4
  $SUSPND, 2–1
    problem calling in cluster environment, 4–62
System time
  changes and enhancements, 3–40
  documentation changes, 3–40

T

Task package, 4–13
TCP/IP
  product requirements, 1–4
  support for MONITOR, 3–28
Terminal Fallback facility (TFF), 3–41
  restrictions, 3–41
TFF
  See Terminal Fallback facility
Thread Independent Services (TIS) interface for DECthreads, 4–26
TMSCP_SERVE_ALL system parameter, 3–44
Token Ring devices
  DECnet support, 1–9
/TRACEBACK option, 4–53
Transaction group
  definition, 3–15
Transports
  specifying in Mail, 2–5
TURBOchannel devices
  booting, 3–35

U

UETP (User Environment Test Package)
  RRD45 CD–ROM testing, 3–42
Upgrading to Version 7.0
  considerations, 1–1
“Uses” clause restriction
  with POLYCENTER Software Installation
    utility, 4–57

V

Variable used as array index, 4–9
VAX 7000 systems
  Dump Off System Disk (DOSD) not supported, 3–43
VAX argument pointers, 4–51
VAX Instruction
  CASE, 4–64
VAX MACRO and Instruction Set Reference Manual
  documentation corrections, 4–64
VAX SDA
  dump file, 3–43
VAXstation installation
  system hangs, 1–6
Vector support, 4–9
VIRTUALPAGECNT, 3–4
VMS$COMMON.DIR file restore problems, 3–8
VMScluster systems, 3–43
  concurrency improvements, 3–44
  FDDI clusters, 3–44
  KFESA/KFESB restrictions, 3–45
  mixed-version clusters, 1–2
  SCSI clusters, 3–43
  system startup problem with
    ALPHAVMSSYS.PAR file, 3–45
  TMSCP_SERVE_ALL parameter, 3–44
VMSINSTAL command procedure
  with layered product installations, 1–5
Volume shadowing
  ACCVIO system failures, 3–47
  documentation corrections
    per-disk licensing, 3–48
  restrictions, 3–46
    DISMOUNT/CLUSTER command, 3–48
    recommended version for KDM70 devices, 3–47
  RZ57 support, 3–48
  SCSI limitation on VAX systems, 3–46
  SHADZEROMBR bugcheck, 3–47
  timer-based polling, 3–46
VT500 series terminals
  baud rates supported, 1–14
  Screen Management (SMG$) Facility support
    for, 4–59

W

Watchpoints, static, 4–4
Windows interface customizations, 4–10
Working set size, 3–2
WSEXTENT qualifier, 3–2
WSMAX, 3–2, 3–4

X

X/Open Transport Interface (XTI), 4–65
XQPCTL2 system parameter, 3–44