HP OpenVMS Version 8.2 Release Notes

Order Number: BA322-90004

January 2005

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 I64 Version 8.2 OpenVMS Alpha Version 8.2

Hewlett-Packard Company
Palo Alto, California

© Copyright 2005 Hewlett-Packard Development Company, L.P.

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

The information contained herein is subject to change without notice. The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

Intel and Itanium are registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Java is a US trademark of Sun Microsystems, Inc.

Oracle is a US registered trademark of Oracle Corporation, Redwood City, California.

OSF and Motif are trademarks of The Open Group in the US and other countries.

UNIX is a registered trademark of The Open Group.

Microsoft, Windows, Windows NT, and MS Windows are US registered trademarks of Microsoft Corporation.

X/Open is a registered trademark, and the X device is a trademark of X/Open Company Ltd. in the UK and other countries.

Printed in the US

ZK6674

The HP OpenVMS documentation set is available on CD-ROM.

This document was prepared using DECdocument, Version 3.3-1b.

Contents

Preface xiii

1 OpenVMS Software Installation and Upgrade Release Notes

1.1 HP Software Technical Support Policy 1–1
1.2 General Application Compatibility Statement 1–2
1.3 Obtaining Remedial Kits 1–3
1.4 Networking Options 1–3
1.5 Disk Incompatibility with Older Versions of OpenVMS 1–3
1.6 HP DECwindows Motif for OpenVMS 1–4
1.7 HP SSL: Installing HP SSL V1.2 1–4
1.8 Release Notes for OpenVMS I64 Users 1–5
1.8.1 HP Integrity Server Configurations 1–5
1.8.2 Clearing the System Event Log (SEL) on Integrity Servers 1–5
1.8.3 Firmware for Integrity Servers 1–6
1.8.4 Booting On a Fibre Channel Storage Device 1–7
1.8.5 Booting from the Installation DVD 1–7
1.8.6 Setting Up I64 Systems to Reboot 1–8
1.8.7 HP DECwindows Motif Release Notes 1–8
1.8.7.1 Keyboard Support 1–8
1.8.7.2 Connect Peripheral Devices Prior to Server Startup 1–8
1.8.7.3 Countdown Messages Displayed During Startup 1–8
1.9 Release Notes for OpenVMS Alpha Users 1–8
1.9.1 Firmware for OpenVMS Alpha Version 8.2 1–9
1.9.2 Upgrade Paths 1–9
1.9.3 Error on Upgrade from Version 7.3-1 1–10
1.9.4 Encryption for OpenVMS Alpha: Remove Before Upgrading 1–10
1.9.5 HP DECram V3.*n*: Remove Before Upgrading 1–11
1.9.6 Kerberos V1.0: Remove Before Upgrading 1–11
1.9.7 Converting the LANCP Device Database After Upgrading 1–12
1.9.8 DECnet-Plus Requires a New Version 1–12
1.9.9 SHADOW_MAX_UNIT Default Setting and Memory Usage 1–13

2 OpenVMS Associated Products Release Notes

2.1 Associated Product Support 2–1
2.2 BASIC: V1.5A Required to Create STARLET Library (Alpha Only) 2–1
2.3 CMAP Files Added 2–2
2.4 COBOL: Changes in I/O Run-Time Diagnostics and RMS Special Registers 2–2
2.5 COM for HP OpenVMS (Alpha Only) 2–2
2.5.1 COM for OpenVMS Support 2–2
2.5.2 Registry Access Error with Heavy Load of Applications 2–2
2.6 DECdfs Version 2.4 Required for OpenVMS Version 8.2 2–3

2.7 DECforms Web Connector Version 3.0 (Alpha Only) 2–3
2.8 DEC PL/I: RTL Support for OpenVMS 2–3
2.9 FMS Kits 2–4
2.10 Graphical Configuration Manager (GCM) (Alpha Only) 2–4
2.11 HP DCE RPC for OpenVMS 2–4
2.11.1 DCE RPC Supports FAILSafe IP 2–4
2.11.2 Support for Tuning the Buffer Size of RPC Sockets 2–4
2.11.3 RTI (Remote Task Invocation) RPC (I64 Only) 2–4
2.11.4 Microsoft Lan Manager RPC Not Tested (I64 Only) 2–4
2.11.5 The utc_mulftime Factor Argument Type 2–5
2.11.6 Support for G_FLOAT and IEEE Floating-Point Types 2–5
2.12 HP DECnet-Plus for OpenVMS: X.25 Data Links Not Supported (I64 Only) 2–6
2.13 HP DECram 2–6
2.13.1 DECram Ships With OpenVMS Version 8.2 2–6
2.13.2 Conflict with DECRYPT DCL Command 2–7
2.13.3 Maximum Disk Size for DECram 2–7
2.13.4 DECram Commands and Errors 2–7
2.13.5 DECram and Volume Shadowing 2–7
2.14 HP DECwindows Motif for OpenVMS 2–7
2.14.1 DECwindows Pause Screen Can Fail to Unlock (Alpha Only) 2–8
2.14.2 New Locales Added 2–8
2.14.3 Available Language Variants 2–8
2.14.4 Support for LAT Transport Interface 2–8
2.14.5 User-Written Transports Not Supported 2–9
2.15 HP Secure Web Server Version Support 2–9
2.16 Pascal 2–10
2.16.1 V5.8A (or Later) Required to Create STARLET Library (Alpha Only) 2–10
2.16.2 Installing HP Pascal After an Upgrade (Alpha Only) 2–10
2.17 WEBES and SEA Support on I64 Systems 2–10

3 General User Release Notes

3.1 OpenVMS Freeware CD–ROMs 3–1
3.1.1 Freeware Menu Unavailable (I64 Only) 3–1
3.2 Online Help Changes 3–2
3.3 DCL Commands 3–2
3.3.1 ANALYZE/ERROR_LOG Command Not Ported to OpenVMS I64 3–2
3.3.2 CREATE/MAILBOX: Temporary Restriction 3–2
3.3.3 DIAGNOSE Command No Longer Supported 3–2
3.3.4 SHOW LICENSE /OE and /HIERARCHY Require SYSLCK (I64 Only) 3–3
3.4 DECmigrate Not On Open Source Tools CD-ROM 3–3
3.5 iconv Fixes 3–3
3.6 HP Secure Web Browser 3–3
3.6.1 Increased Memory Required 3–3
3.6.2 Installation Error on ODS-2 Disk Volume (I64 Only) 3–4
3.7 TECO Editor Not Yet Available on I64 Systems 3–4

4 System Management Release Notes

4.1 Recovering from System Hangs or Crashes (I64 Only) 4–1
4.2 AUTHORIZE: New Quotas for the DEFAULT and SYSTEM Account 4–1
4.3 AUTOGEN: New Behavior with NEWPARAMS.DAT Files 4–2
4.4 Backup Utility: Behavior Change 4–2
4.5 DECdtm/XA with Oracle® 8*i* and 9*i* (Alpha Only) 4–2
4.6 Device Unit Number Maximum Increased 4–3
4.7 ECP Data Collector and Performance Analyzer V5.5 (Alpha Only) 4–3
4.8 EDIT/FDL: Fixing Recommended Bucket Size 4–3
4.9 EFI$CP Utility: Use Not Recommended 4–3
4.10 EFI Tools: VMS_SHOW DUMP_DEV Errors (I64 Only) 4–4
4.11 Error Log Viewer (ELV) Utility: TRANSLATE/PAGE Command 4–4
4.12 External Authentication 4–4
4.12.1 I64 External Authentication Support 4–4
4.12.2 SET PASSWORD Behavior Within a DECterm Terminal Session 4–4
4.12.3 No Password Expiration Notification on Workstations 4–5
4.13 INITIALIZE Command Changes 4–5
4.14 INSTALL Utility 4–6
4.15 Lock Manager 4–6
4.15.1 Lock Value Block Extended 4–6
4.15.2 Fast Lock Remastering and PE1 4–6
4.16 Logical Disk (LD) Utility: Problem Fixed 4–7
4.17 On-Disk Structure (ODS) Layout Changes 4–7
4.18 OpenVMS Cluster Systems 4–8
4.18.1 OpenVMS I64 Cluster Support 4–8
4.18.2 Temporary Exceptions 4–8
4.18.3 Permanent Exceptions 4–8
4.18.4 Patch Kits Needed for Cluster Compatibility 4–9
4.18.5 New API Can Correct Incompatibility of Fibre Channel and SCSI Multipath with Some Third-Party Products 4–10
4.18.6 CLUSTER_CONFIG.COM and Limits on Root Directory Names 4–11
4.18.7 Cluster Performance Reduced with CI-LAN Circuit Switching 4–11
4.18.8 Gigabit Ethernet Switch Restriction in an OpenVMS Cluster System 4–12
4.18.9 Multipath Tape Failover Restriction 4–13
4.18.10 No Automatic Failover for SCSI Multipath Medium Changers 4–13
4.19 OpenVMS Galaxy (Alpha Only) 4–13
4.19.1 Galaxy Definitions 4–14
4.19.2 OpenVMS Graphical Configuration Manager 4–14
4.19.3 Galaxy on ES40: Uncompressed Dump Limitation 4–14
4.19.4 Galaxy on ES40: Turning Off Fast Path 4–14
4.20 OpenVMS Management Station 4–15
4.21 OpenVMS Registry Can Corrupt Version 2 Format Database 4–15
4.22 Security: Changes to DIRECTORY Command Output 4–15
4.23 Server Management Process (SMHANDLER) 4–15
4.24 SYSGEN: Security Auditing Fixed 4–16
4.25 SYSMAN Utility: DUMP_PRIORITY Command Changes 4–16
4.26 System Parameters 4–16
4.26.1 New System Parameters 4–17
4.26.2 Obsolete System Parameters 4–17
4.26.3 Displaying Obsolete System Parameters 4–17
4.26.4 System Parameter Changes 4–18
4.26.5 MMG_CTLFLAGS: Documentation Errors 4–20
4.27 Terminal Fallback Facility (TFF) 4–20

4.28 Time Zone Changes 4–21
4.29 User Environment Test Package (UETP) (I64 Only) 4–22
4.30 Recommended Caching Methods 4–22
4.31 Volume Shadowing for OpenVMS 4–22
4.31.1 Device Name Requirement 4–22
4.31.2 Warning About Using SET SHADOW and SHOW SHADOW in DCL Command Procedures 4–23
4.31.3 Write Bitmaps and Dissimilar Device Shadowing (DDS) Caution 4–23
4.31.4 KZPDC (Smart Array 5300) Restrictions 4–24
4.31.5 Changes in Shadow Set Merge Delay Computation 4–25
4.31.6 ANALYZE/DISK/SHADOW Command Behavior 4–25
4.31.7 ANALYZE/DISK/SHADOW Command Behavior with Dissimilar Device Shadow Sets 4–26
4.31.8 Dismount of Shadow Set Member Using /MINICOPY 4–26

5 Programming Release Notes

5.1 Programs Must Be Recompiled and Linked (I64 Only) 5–1
5.2 Privileged Programs May Need to Be Recompiled (Alpha Only) 5–1
5.3 Privileged Data Structures Updates 5–1
5.3.1 KPB Extensions 5–2
5.3.2 CPU Name Space 5–2
5.3.3 64-Bit Logical Block Number (LBN) 5–2
5.3.4 Forking to a Dynamic Spinlock 5–3
5.3.5 UCB/DDB Updates 5–3
5.3.6 PCB$T_TERMINAL Size Increase 5–3
5.3.7 Per-Thread Security Impacts Privileged Code and Device Drivers 5–4
5.3.8 IPL Requirement For OpenVMS Fork Thread Creation Now Enforced 5–5
5.4 Applications Using Floating-Point Data 5–5
5.5 Ada Compiler Not Yet Available (I64 Only) 5–6
5.6 Backup API: Journaling Callback Events Restriction 5–6
5.7 C Programs: Compiling with CASE_LOOKUP=SENSITIVE Settings 5–6
5.8 C Run-Time Library 5–6
5.8.1 Memory Leak in Programs Using socket_fd Fixed 5–7
5.8.2 vsnprintf and snprintf User Buffer Overwrite Fixed 5–7
5.8.3 mmap and mprotect Changes 5–7
5.8.4 getpwnam_r and getpwuid_r Pointer Problem Fixed 5–7
5.8.5 _strtok_r32 and _strtok_r64 Now in Scope 5–7
5.8.6 const Type Qualifier Added to iconv Prototype (Alpha Only) 5–8
5.8.7 <stddef.h> Header Modified (Alpha Only) 5–8
5.8.8 getc Macro Argument Now Protected by Parentheses (Alpha Only) . . . 5–8
5.8.9 CXXL Prefix Problems with Inlined Functions getc and getchar Fixed (Alpha Only) 5–8
5.8.10 Non std Functions Declared in std Namespace (Alpha Only) 5–8
5.8.11 lseek on Large File Offset Problem Fixed (Alpha Only) 5–9
5.8.12 New EABANDONED Code in <errno.h> 5–9
5.8.13 mktime Problem Fixed 5–9
5.8.14 POLLWRNORM Now the Same as POLLOUT in <poll.h> 5–9
5.8.15 IPV6 Structures in <in6.h> Now Packed 5–9
5.8.16 __PAL_BUGCHK Fixed in <builtins.h> 5–9
5.8.17 C++ Compiler Error with statvfs Fixed 5–9
5.8.18 glob and globfree Issues Fixed 5–10
5.8.19 DECC$SHR_EV56 Now Linked Correctly 5–10

5.8.20 Zone Information Compiler (zic) Updates 5–10
5.9 Calling Standard and Rotating Registers (I64 Only) 5–10
5.10 Common Data Security Architecture (CDSA) Considerations 5–11
5.10.1 Secure Delivery Advanced Developer's Kit 5–11
5.10.2 Installation and Initialization Considerations 5–11
5.11 CONVERT-I-SEQ Error on CONVERT/NOSORT with Collated Key 5–12
5.12 Debugging Modes: Avoiding CPUSPINWAIT Bugchecks 5–12
5.13 Delta/XDelta Debuggers 5–13
5.13.1 Delta Debugger Not Available on OpenVMS I64 5–13
5.13.2 Restrictions for XDELTA on I64 Systems 5–13
5.13.3 Differences Between XDELTA on I64 Systems and Alpha Systems . . . 5–13
5.13.4 XDELTA Register Display Consideration (I64 only) 5–13
5.14 File Applications: Corrections to *Guide to OpenVMS File Applications* 5–14
5.15 HP BLISS Compiler Warnings with RMS Structures (I64 Only) 5–14
5.16 HP COBOL Run-Time Library (RTL) 5–15
5.16.1 SET and COB$SWITCHES 5–15
5.16.2 Record Locking Problem Corrected 5–15
5.17 HP Decimal Support Run-Time Library (RTL) 5–15
5.18 HP Fortran for I64 5–16
5.19 HP MACRO for OpenVMS 5–16
5.19.1 HP MACRO for OpenVMS I64 5–17
5.19.2 /TIE Qualifier Defaults Differ on Alpha and I64 5–17
5.19.3 /OPTIMIZE=VAXREGS Qualifier Not Supported on I64 5–17
5.19.4 CODGENWARN Message Can Be Ignored (Alpha Only) 5–17
5.19.5 Granularity Support (I64 Only) 5–17
5.19.6 Integer Divide-by-Zero Error Not Raised By Default (I64 Only) 5–17
5.19.7 Integer Divide By Most Negative Number Crashes Compiler (I64 Only) 5–18
5.19.8 Integer Divide Might Not Set ''V'' Condition Code Properly (I64 Only) 5–18
5.19.9 Floating Divide-by-Zero Error Not Raised (I64 Only) 5–18
5.19.10 INSV Instruction Overwrites Extra Memory (I64 Only) 5–18
5.20 Hypersort Utility 5–19
5.20.1 Reporting a Problem to HP 5–19
5.20.2 Large Files Restriction 5–19
5.20.3 Hypersort and VFC Files Restriction 5–19
5.20.4 /FORMAT=RECORD_SIZE Restriction 5–19
5.20.5 Using Hypersort with Search Lists and Other Uses of Logical Names 5–20
5.20.6 Lack of Free Space for Work Files 5–20
5.20.7 Input Asterisk (*) Restriction 5–20
5.20.8 Optimal Working Set Extent and Page File Quota Settings 5–20
5.21 Intel® Assembler (I64 Only) 5–20
5.22 Librarian Utility 5–20
5.22.1 Linking Against Data-Reduced ELF Object Libraries Not Recommended (I64 Only) 5–20
5.22.2 Failure to Insert or Replace .STB files in an I64 Library (I64 Only) 5–21
5.22.3 Librarian Fails to Report Errors When Process Quota Too Low 5–21
5.22.4 Object Module Name Length Problem (I64 Only) 5–21
5.23 Lightweight Directory Access Protocol (LDAP) API 5–22
5.23.1 The Routine ldap_get_option Returns Error -1 When ld Is NULL 5–22
5.23.2 The Routine ber_flatten( ) Fails to Detect Mismatched Braces 5–22

5.24 Linker Utility for OpenVMS Alpha 5–22
5.24.1 LINK/NATIVE_ONLY Help Text Clarification 5–22
5.24.2 Linker Appears to Hang When Many Files Are Specified 5–22
5.24.3 Change in Linker Default Behavior with Library Check 5–24
5.24.4 Limit of 25 Elements on Stack 5–24
5.25 Linker Utility for OpenVMS I64 5–24
5.25.1 Differences Between the I64 and Alpha Linkers 5–24
5.25.2 LINK_ORDER Section Header Flag Not Supported 5–24
5.25.3 Linking Against Data-Reduced ELF Object Libraries Not Recommended 5–25
5.25.4 Errors in the Image Synopsis Section of the Map 5–25
5.25.5 DIFTYPE and RELODIFTYPE Messages 5–25
5.26 LTDRIVER: CANCEL SELECTIVE Restriction 5–27
5.27 Mail Utility: Threads Restriction for Callable Mail 5–27
5.28 OpenVMS Debugger 5–27
5.28.1 General Conditions and Workarounds (I64 Only) 5–28
5.28.2 BASIC Language Issues (I64 Only) 5–28
5.28.3 C++ Language Issues (I64 Only) 5–28
5.28.4 COBOL Language Issues (I64 Only) 5–29
5.28.5 Fortran Language Issues (I64 Only) 5–29
5.28.6 Pascal Language Issues (I64 Only) 5–29
5.28.7 SET SCOPE Command: Behavior Change 5–29
5.28.8 SHOW IMAGE Command Change 5–29
5.28.9 Client/Server Interface: Previous Versions Not Supported (Alpha Only) 5–29
5.29 OpenVMS System Dump Analyzer (SDA) 5–30
5.29.1 READ Command Default Change 5–30
5.29.2 CLUE Commands Not Ported to OpenVMS I64 5–30
5.29.3 SHOW CALL_FRAME Functionality Changes (I64 Only) 5–30
5.30 PL/I Libraries Not Included in OpenVMS I64 Version 8.2 5–31
5.31 POSIX Threads Library 5–31
5.31.1 Stack Overflows During Exception Handling (I64 Only) 5–31
5.31.2 THREADCP Command Behavior on I64 Systems 5–32
5.31.3 Floating-Point Compilations and Exceptions (I64 Only) 5–32
5.31.4 C Language Compilation Header File Changes 5–32
5.31.5 New Priority Adjustment Algorithm 5–33
5.31.6 Process Dumps 5–33
5.31.7 Dynamic CPU Configuration Changes 5–34
5.31.8 Debugger Metering Function Does Not Work 5–34
5.32 RMS: Improved Global Buffer VA Management 5–34
5.33 RMS Journaling 5–34
5.33.1 Recovery Unit Journaling Incompatible with Kernel Threads 5–35
5.33.2 Modified Journal File Creation 5–35
5.33.3 Remote Access of Recovery Unit Journaled Files in an OSI Environment 5–35
5.33.4 After-Image (AI) Journaling 5–35
5.33.5 VFC Format Sequential Files 5–36
5.34 RTL Library (LIB$) 5–36
5.34.1 RTL Library (LIB$) Help Omission 5–36
5.34.2 RTL Library (LIB$): Calling Standard Routines (I64 Only) 5–36
5.35 Screen Management (SMG$) Documentation 5–37
5.36 SORT32 Utility 5–38
5.36.1 CONVERT Problem With DFS-Served Disks 5–38
5.36.2 Temporary Work Files Not Always Deleted 5–38

5.36.3 SORT/SPECIFICATION With Compound Conditions: Requirement . . . 5–38
5.36.4 Performance Problem with Variable Length Records . . . 5–39
5.36.5 Work File Directories Restriction . . . 5–39
5.37 System Code Debugger (SCD) Not Available on I64 Systems . . . 5–39
5.38 System Services . . . 5–39
5.38.1 $GETJPI Item Code SCHED_CLASS_NAME Documented Incorrectly . . . 5–39
5.38.2 New GETSYI Item Codes Required for PFN-Mapped Sections (I64 Only) . . . 5–39
5.38.3 PFN-Mapped Sections and Uncached Memory (I64 Only) . . . 5–40
5.38.4 SYS$ACM: Using on I64 Systems . . . 5–40
5.38.5 SYS$GOTO_UNWIND Not Available on I64 . . . 5–41
5.39 Timer Queue Entries (TQEs) . . . 5–41
5.40 Traceback Facility . . . 5–41
5.40.1 API Error (I64 Only) . . . 5–41
5.40.2 Problem Corrected . . . 5–41
5.41 Watchpoint Utility (I64 Only) . . . 5–42
5.42 Whole Program Floating-Point Mode (I64 Only) . . . 5–42

6 Hardware Release Notes

6.1 MP and BMC Console Restrictions (I64 Only) . . . 6–1
6.1.1 Input, Output, and Error Device Restriction . . . 6–1
6.1.2 Remapping Ctrl/H to the Delete Key . . . 6–2
6.2 AlphaServer 2100 . . . 6–2
6.2.1 Console Display . . . 6–2
6.2.2 SCSI Controller Restriction . . . 6–3
6.3 AlphaServer 8200/8400: FRU Table Error . . . 6–3
6.4 AlphaServer ES47/ES80/GS1280 Systems . . . 6–3
6.4.1 INIT Console Command Usage on ES47/ES80/GS1280 Soft Partitions . . . 6–3
6.4.2 Setting SYSGEN Parameter PHYSICAL_MEMORY . . . 6–3
6.4.3 RAD Support . . . 6–4
6.4.4 License Requirements . . . 6–4
6.4.5 STOP/CPU and Shutdown Behavior . . . 6–4
6.4.6 Setting Time at MBM . . . 6–4
6.5 AlphaServer GS Series Systems . . . 6–4
6.5.1 AlphaServer GS80/160/320 Systems: Device Restriction . . . 6–4
6.5.2 OpenVMS Galaxy License Enforcement . . . 6–5
6.5.3 Installing Licenses . . . 6–5
6.5.4 AlphaServer GS60/GS60E/GS140 Multiple I/O Port Module Configuration Restriction . . . 6–7
6.6 AlphaStation 200/400: ISA_CONFIG.DAT Changes Required . . . 6–7
6.7 AlphaStation 255: PCI Configuration Restriction . . . 6–8
6.8 ATI RADEON 7000 Graphics (I64 Only) . . . 6–8
6.8.1 I64 Graphics Support . . . 6–8
6.8.2 Hardware Accelerated 3D Graphics Not Supported on RADEON 7000 . . . 6–8
6.9 ATI RADEON 7500 Graphics . . . 6–8
6.9.1 Resource Requirements . . . 6–9

6.9.2 New Features in Version 8.2 6–9
6.9.2.1 Improved Performance of Text and Other Bitmap Operations 6–9
6.9.2.2 Improved Performance of DMA Operations 6–9
6.9.2.3 Improved Performance of OpenGL Texture Copies 6–10
6.9.2.4 Screen Resolution 1024x864 Added 6–10
6.9.2.5 Hardware-Accelerated Indirect 3D Rendering 6–10
6.9.3 Problems Fixed (Alpha Only) 6–10
6.9.4 DECW$OPENGLSHR_RADEON.EXE Renamed to DECW$MESA3DSHR_RADEON.EXE 6–11
6.9.5 Support Limitations 6–11
6.9.6 Video Artifacts at High Refresh Rates 6–11
6.9.7 OpenGL Supports IEEE Arithmetic Only 6–11
6.9.8 DECwindows Server Hangs When Output Is Written to the Graphics Console (OPA0) 6–12
6.9.9 Monitor Must Be Connected During Initialization 6–12
6.9.10 Boot Reset Recommendation (Alpha Only) 6–12
6.9.11 No Overlay Planes 6–12
6.9.12 Single Colormap 6–12
6.9.13 Single Bit Depth for All Windows 6–13
6.9.14 Pixel Depth for Read/Write Color Map 6–13
6.9.15 Backing Store/Save Unders Not Supported for 3D Windows 6–13
6.9.16 Threads Restriction 6–13
6.9.17 No Support for Single Buffered Visuals 6–13
6.9.18 No 3D Support for Color Index Mode 6–13
6.9.19 Timer Mechanism 6–14
6.10 DECwindows X11 Display Server (Alpha Only) 6–14
6.10.1 Vendor and Operating System Version Data Has Been Corrected 6–14
6.10.2 S3 Multihead Graphics 6–14
6.10.3 Integrated Graphics Board Support 6–15
6.11 DIGITAL Modular Computing Components (DMCC) 6–15
6.11.1 Alpha 5/366 and 5/433 PICMG SBC Restriction 6–15
6.11.2 Updating the SRM Console 6–15
6.12 Digital Personal Workstation: Booting OpenVMS V7.3-1 and Higher 6–15
6.13 Dual-Controller HSG*nn* with Many LUNs Can Fail Under Heavy I/O Load 6–16
6.14 Open3D Graphics Licensing Change 6–16
6.15 PowerStorm 300/350 PCI Graphics Support for OpenVMS 6–17
6.16 RF*nn* DSSI Disk Devices and Controller Memory Errors 6–17
6.17 RZ*nn* Disk Drive Considerations 6–20
6.17.1 RZ25M and RZ26N Disk Drives: Recommendations 6–20
6.17.2 RZ26N and RZ28M Disks: Recommended Firmware Support 6–20
6.17.3 RZ26L and RZ28 Disks: Required Firmware for Multihost Use 6–20
6.17.3.1 Firmware Revision Level 442 Requirements 6–21
6.17.3.2 Firmware Revision Level 442 Installation Procedure 6–21
6.18 ZLX Graphics Boards Support 6–21
6.19 Recompiling and Relinking OpenVMS Device Drivers 6–22
6.19.1 Alpha and VAX SCSI Device Drivers 6–22
6.19.2 OpenVMS Alpha Device Drivers 6–22
6.20 Device Driver MON Version Handling 6–22
6.21 Possible Per-Threads Security Impact on Alpha Device Drivers 6–22
6.22 Device IPL Setup for OpenVMS Alpha Drivers 6–23
6.23 CRCTX Routines Enhanced 6–23

A Product Retirement Notices

A.1 Compaq Open3D Layered Product Not Supported in Version 8.2 . . . . . A–1
A.2 Open3D Graphics Licensing Change A–2
A.3 DECamds Not Supported on OpenVMS Version 8.2 . . . . . A–2
A.4 DECevent Not Supported A–2
A.5 DECwrite Reached End-of-Service Life in December 2004 . . . . . A–2
A.6 Error Log Report Formatter (ERF) Unsupported A–3
A.7 ISA_CONFIG.DAT Unsupported in Future Release A–3
A.8 POSIX 1003.4a Draft 4 Interface May Be Retired A–3
A.9 Visual Threads: Not Supported for Version 8.2 . . . . . A–3
A.10 Last Ordering Dates for Alpha Systems A–4
A.11 Archived Manuals A–4

B Interlocked Memory Instructions (Alpha Only)

B.1 Required Code Checks B–1
B.2 Using the Code Analysis Tool (SRM_CHECK) B–1
B.3 Noncompliant Code Characteristics B–2
B.4 Coding Requirements B–3
B.5 Compiler Versions B–5
B.6 Recompiling Code with ALONONPAGED_INLINE or LAL_REMOVE_FIRST B–6

Index

Tables

1–1 Supported Versions of DECwindows Motif 1–4
4–1 Patch Kits Required for Cluster Compatibility 4–9
4–2 Galaxy Definitions 4–14
4–3 TFF Character Fallback Tables 4–21
6–1 Changes to Device Description Block 6–7
6–2 Supported Microcode Revision Levels 6–18
6–3 Commands for Updating Microcode in Certain DSSI Disk Devices . . . 6–19
6–4 Revision Level 442 Firmware Compatibility 6–21
B–1 Versions of OpenVMS Compilers B–5

Preface

Intended Audience

This manual is intended for all users of the HP OpenVMS Alpha or HP OpenVMS I64 Version 8.2 operating system. Read this manual before you install, upgrade, or use OpenVMS Version 8.2.

Document Structure

This manual contains the following chapters and appendixes:

- Chapter 1 contains release notes that pertain to upgrading or installing the OpenVMS Alpha operating system or installing OpenVMS I64.
- Chapter 2 contains installation and support information for OpenVMS associated products.
- Chapter 3 contains release notes about the general use of the OpenVMS operating system.
- Chapter 4 contains release notes specific to OpenVMS system management.
- Chapter 5 contains release notes that relate to programming on an OpenVMS system, including notes for compilers, linkers, and run-time library routines.
- Chapter 6 contains information pertaining to hardware that runs on the OpenVMS operating system and OpenVMS device support.
- Appendix A contains information about OpenVMS products that are no longer supported as of this release, or that are slated for retirement.
- Appendix B describes the proper use of interlocked memory instructions, which is crucial for the Alpha 21264 (EV6) processor.

Notes are organized by facility or product name when applicable.

This manual contains release notes introduced in the current release and notes from previous OpenVMS Alpha versions that still apply to the new release. A subheading for each release note indicates either the version of origin (for example, *V7.3-2*) or the version when an old note was last updated (for example, a note from V7.3-2 that was revised for Version 8.2 will be labeled with *V8.2*).

Notes from previous releases are published when:

- The information in the release note has not been documented in any other printed 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 *HP OpenVMS Version 8.2 New Features and Documentation Overview* manual.

For additional information about HP OpenVMS products and services, visit the following World Wide Web address:

`http://www.hp.com/go/openvms`

Reader's Comments

HP welcomes your comments on this manual. Please send comments to either of the following addresses:

Internet	**openvmsdoc@hp.com**
Postal Mail	Hewlett-Packard Company OSSG Documentation Group, ZKO3-4/U08 110 Spit Brook Rd. Nashua, NH 03062-2698

How to Order Additional Documentation

For information about how to order additional documentation, visit the following World Wide Web address:

`http://www.hp.com/go/openvms/doc/order`

Conventions

The following conventions may be 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*	A sequence such as PF1 *x* indicates that you must first press and release the key labeled PF1 and then press and release another key or a pointing device button.
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.) In the HTML version of this document, this convention appears as brackets, rather than a box.
. . .	A horizontal ellipsis in examples indicates 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.
. . .	A vertical ellipsis indicates 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 you must enclose choices in parentheses if you specify more than one.
[ ]	In command format descriptions, brackets indicate optional choices. You can choose one or more items or no items. Do not type the brackets on the command line. However, you must include the brackets in the syntax for OpenVMS directory specifications and for a substring specification in an assignment statement.
|	In command format descriptions, vertical bars separate choices within brackets or braces. Within brackets, the choices are optional; within braces, at least one choice is required. Do not type the vertical bars on the command line.
{ }	In command format descriptions, braces indicate required choices; you must choose at least one of the items listed. Do not type the braces on the command line.
bold type	Bold type represents the introduction of a new term. It also represents the name of an argument, an attribute, or a reason.
italic type	Italic type 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 *dd* represents the predefined code for the device type).
UPPERCASE TYPE	Uppercase type indicates a command, the name of a routine, the name of a file, or the abbreviation for a system privilege.
`Example`	This typeface indicates code examples, command examples, and interactive screen displays. In text, this type also identifies URLs, UNIX commands and pathnames, PC-based commands and folders, and certain elements of the C programming language.
-	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 Software Installation and Upgrade Release Notes

This chapter contains information that you need to know before installing OpenVMS I64 Version 8.2 or installing or upgrading to OpenVMS Alpha Version 8.2. Topics of interest to both Alpha and I64 users are covered first. Later sections group notes of interest to users of specific platforms.

HP recommends that you read all of the following manuals before installing or upgrading OpenVMS Version 8.2:

- *HP OpenVMS Version 8.2 Release Notes* (this manual)
- *HP OpenVMS Version 8.2 New Features and Documentation Overview*
- *HP OpenVMS Version 8.2 Upgrade and Installation Manual*

Refer to Chapter 6 for hardware release notes and to Chapter 2 for information about associated products.

1.1 HP Software Technical Support Policy

HP provides software technical support for OpenVMS operating system software for the latest, currently shipping version and the immediate prior version of the product. Each version is supported for 24 months from its release date, or until the release of the second subsequent version, whichever is greater. ''Version'' is defined as a release containing new features and enhancements. Patch kits and maintenance-only releases do not meet the definition of ''version'' in the context of this support policy.

Current version-level support (Standard Support or SS) and Prior Version Support (PVS) for OpenVMS operating system software is provided for OpenVMS versions in accordance with these guidelines. The current level of support for recent versions of OpenVMS Alpha and OpenVMS VAX is kept up to date on this web page:

`http://h71000.www7.hp.com/openvms/openvms_supportchart.html`

The Operating System Support Policy applies to all OpenVMS Major Releases, New Feature Releases, and Enhancement releases, which are defined as follows:

- **Major Releases** for OpenVMS contain substantial new functionality. The version number increases to the next integer (for example, from 6.2-1H1 to 7.0).

  Application impact: OpenVMS internal interfaces have changed. Although binary compatibility will be maintained for the majority of applications, independent software vendors (ISVs) should test on the new version and may need to release a new application kit. Some application partners may want to release a new application kit to take advantage of new features in the operating system.

- **New Feature Releases** for OpenVMS contain new features, enhancements, and maintenance updates. The version number increases to the next decimal number (for example, from 7.2 to 7.3).

  Application impact: The release has not retired any published application programming interfaces (APIs). However, OpenVMS internal interfaces may have been modified with the addition of significant new functionality or implementation of performance improvements. It is unlikely that a new application kit will be required for 95 percent of all applications that use documented APIs. Device driver and kernel-level applications (that is, those that use nonstandard or undocumented APIs) may need qualification testing.

- **Enhancement Releases** for OpenVMS contain enhancements to existing features and maintenance updates. The version number increases to show a revision by using a dash (for example, OpenVMS Version 7.3-2).

  Application impact: The release may contain new hardware support, software enhancements, and maintenance, but the changes are isolated and have no impact on applications that use published APIs. There is no need for ISVs to test on the new release or to produce a new application kit.

- **Hardware Releases** provide current version-level support until 12 months after a subsequent release containing that particular hardware support. Hardware releases are shipped with new hardware sales only and are not distributed to existing service customers.

The following OpenVMS core products are supported at the same level (Standard Support or Prior Version Support) and duration as the operating system version on which they ship:

- HP Advanced Server for OpenVMS
- HP DECnet (Phase IV)
- HP DECnet-Plus for OpenVMS
- HP OpenVMS Cluster Client Software
- HP OpenVMS Cluster Software for OpenVMS
- HP PathWorks or HP PATHWORKS for OpenVMS
- HP RMS Journaling for OpenVMS
- HP TCP/IP Services for OpenVMS
- HP Volume Shadowing for OpenVMS

These products require their own individual support contracts and are not included in the operating system support contract.

1.2 General Application Compatibility Statement

OpenVMS has consistently held the policy that published APIs are supported on all subsequent releases. It is unlikely applications that use published APIs will require changes to support a new release of OpenVMS. APIs may be "retired," and thus removed from the documentation; however, the API will continue to be available on OpenVMS as an undocumented interface.

1.3 Obtaining Remedial Kits

Remedial kits for HP products are available online at the HP IT Resource Center (ITRC). Use of the ITRC patch download site requires user registration and login. Registration is open to all users and no service contract is required. You can register and log in from the following URL:

`http://www2.itrc.hp.com/service/patch/mainPage.do`

You can also use FTP to access patches from the following location:

`ftp://ftp.itrc.hp.com/openvms_patches`

1.4 Networking Options

V8.2

OpenVMS provides customers with the flexibility to choose the network protocol of their choice. Whether you require DECnet or TCP/IP, OpenVMS allows you to choose the protocol or combination of protocols that works best for your network. OpenVMS can operate with both HP and third-party networking products.

During the main installation procedure for OpenVMS Version 8.2, you have the option of installing the following supported HP networking software:

- Either HP DECnet-Plus Version 8.2 for OpenVMS or HP DECnet Phase IV Version 8.2 for OpenVMS. (Note that both DECnet products cannot run concurrently on your system.)

  DECnet-Plus contains all the functionality of the DECnet Phase IV product, plus the ability to run DECnet over TCP/IP or OSI protocols.

  Support for DECnet Phase IV is provided to customers with a Prior Version Support Contract. For more information about the Prior Version Support service, see Section 1.1.

- HP TCP/IP Services for OpenVMS Version 5.5

  TCP/IP Services and DECnet can run concurrently on your system. Once you have installed HP DECnet-Plus for OpenVMS and TCP/IP Services on your system, you can run DECnet applications and OSI applications, or both, over your TCP/IP network. Refer to the *DECnet-Plus for OpenVMS Management Guide* for more information about running DECnet over TCP/IP (RFC 1859) and OSI over TCP/IP (RFC 1006).

Alternatively, after you install OpenVMS, you can install your choice of another third-party networking product that runs on OpenVMS Version 8.2.

For information about how to configure and manage your HP networking software after installation, refer to the TCP/IP, DECnet-Plus, or DECnet documentation. The manuals are available in online format on the OpenVMS Documentation CD-ROM; printed manuals can be ordered from HP (800-282-6672).

1.5 Disk Incompatibility with Older Versions of OpenVMS

V8.2

The OpenVMS Version 8.2 installation procedure initializes the target disk with volume expansion (INITIALIZE/LIMIT). This renders the disk incompatible with versions of OpenVMS prior to Version 7.2. In most cases, this does not present a problem. However, if you intend to mount this new disk on a version of OpenVMS prior to Version 7.2, you must first take steps to make the disk compatible for

that operating system version. Refer to the *HP OpenVMS Version 8.2 Upgrade and Installation Manual* for detailed instructions.

Note that by taking these steps, your new system disk might include a relatively large minimum allocation size (as defined by /CLUSTER_SIZE). As a result, small files will use more space than necessary. Therefore, you should perform these steps *only* for system disks that must be mounted on versions of OpenVMS prior to Version 7.2.

Note

ODS-5 disks are also incompatible with versions of OpenVMS prior to Version 7.2.

1.6 HP DECwindows Motif for OpenVMS

V8.2

The following table lists which versions of DECwindows Motif are supported on various platforms of the OpenVMS Version 8.2 operating system.

Table 1–1 Supported Versions of DECwindows Motif

OpenVMS Version	DECwindows Motif Version
OpenVMS I64 8.2	DECwindows Motif for OpenVMS I64 V1.5
OpenVMS Alpha 8.2	DECwindows Motif for OpenVMS Alpha V1.5

The DECwindows Motif software relies on specific versions of OpenVMS server and device driver images. Be sure to install or upgrade to the version of DECwindows Motif that is appropriate to your operating system environment, as noted in Table 1–1.

For information on support for prior versions of DECwindows Motif, see the *HP DECwindows Motif for OpenVMS Release Notes*.

For detailed information about installing the DECwindows Motif software, refer to the *HP DECwindows Motif for OpenVMS Installation Guide*.

1.7 HP SSL: Installing HP SSL V1.2

V8.2

Installing HP SSL V1.2 on your Alpha or I64 system provides you with new ciphers and algorithms as well as the latest security fixes from openssl.org. The HP SSL V1.2 kit is included with the layered products distribution. HP recommends that you install HP SSL V1.2 after you install or upgrade your operating system and TCP/IP Services for OpenVMS.

You must manually remove any previous versions of SSL before installing SSL V1.2. Use the following commands to remove and install SSL:

```
$ PRODUCT REMOVE SSL
$ PRODUCT INSTALL SSL /SOURCE=ddcu:[directory]
```

For details about installing and using SSL, refer to the *HP Open Source Security for OpenVMS, Volume 2: HP SSL for OpenVMS* manual.

1.8 Release Notes for OpenVMS I64 Users

The following notes are primarily of interest to users of OpenVMS I64 systems.

1.8.1 HP Integrity Server Configurations

V8.2

The OpenVMS I64 Version 8.2 release supports all of the standard system features and core I/O on the rx1600-2, rx1620-2, rx2600-2, rx2620-2, and rx4640-8 Integrity servers.

For OpenVMS I64 Version 8.2, there are two primary storage interconnects:

- The SCSI storage type is U320, used for core I/O for certain Integrity server systems, as well as the A7173A U320 SCSI adapter. For connection to external SCSI storage, the supported storage shelves are the DS2100 or the MSA30.
- The external Fibre Channel storage connection is through the dual-port 2GB Fibre Channel Universal PCI-X adapter (A6826A). This adapter allows for connectivity to any external SAN-based Fibre Channel storage infrastructure supported by OpenVMS.

Customers who used any earlier evaluation or field test kits should note the following important considerations:

- In the SCSI adapter area, the U160 adapter (A6829A) is not officially supported on OpenVMS I64 Version 8.2, and will reach end-of-life during 2005. However, you can use this adapter for existing hardware configurations as long as the system remains as it is currently configured. Any additional adapters, or movement to another server, require the U320 SCSI adapter technology.
- In the case of Fibre Channel, customers might have been running with the AB232A or KGPSA-EA FC adapters. These adapters are not supported on OpenVMS I64 Version 8.2, and customers using them must upgrade to the A6826A FC adapter before running production applications on Version 8.2.

1.8.2 Clearing the System Event Log (SEL) on Integrity Servers

V8.2

HP Integrity servers maintain a System Event Log (SEL) within system console storage, and OpenVMS I64 automatically transfers the contents of the SEL into the OpenVMS error log. If you are operating from the console during a successful boot operation, you might see a message indicating that the Baseboard Management Controller (BMC) SEL is full. You can safely continue when the BMC SEL is full by following the prompts; OpenVMS will process the contents of the SEL. If you wish to clear the SEL manually, enter the following command at the EFI Shell prompt:

```
Shell> clearlogs SEL
```

This command deletes the contents of the SEL. The command is available with current system firmware versions.

If your Integrity server is configured with a Management Processor (MP) and you see a BMC event log warning while connected to the MP console, you can clear the BMC event log by using MP. Press Ctrl/B to drop to the MP> prompt. At the MP> prompt, enter SL (from the main menu) and use the C option to clear the log.

HP recommends that you load and use the most current system firmware. For more information about updating the system firmware, refer to the *HP OpenVMS Version 8.2 Upgrade and Installation Manual*.

1.8.3 Firmware for Integrity Servers

V8.2

OpenVMS I64 Version 8.2 was tested with the latest firmware for each of the supported Integrity servers. The following table lists the recommended firmware versions:

System	System Firmware	BMC Firmware	MP Firmware	DHCP Firmware
rx1600-2	1.10	2.33	E.02.29	N/A
rx1620-2	2.11	3.48	E.03.13	N/A
rx2600-2	2.31	1.52	E.02.29	N/A
rx2620-2	3.10	3.47	E.03.13	N/A
rx4640-8	2.13 3.11	2.35 3.47	E.02.29 E.03.13	1.10 1.10

Some new rx4640 servers require, and will ship with, the higher firmware versions shown in the table. As this manual goes to press, the lower firmware versions are the latest released to customers to update existing servers. OpenVMS has been tested with both versions of firmware. HP recommends that all rx4640 servers be updated to the higher firmware versions when the firmware is released.

To check the firmware versions for your server, use the Extensible Firmware Interface (EFI) command INFO FW, as shown in the following example. (For instructions on how to access and use EFI, refer to the *HP OpenVMS Version 8.2 Upgrade and Installation Manual*.)

```
Shell> INFO FW

FIRMWARE INFORMATION

   Firmware Revision: 2.13 [4412]          ❶

   PAL_A Revision: 7.31/5.37
   PAL_B Revision: 5.65
   HI Revision: 1.02

   SAL Spec Revision: 3.01
   SAL_A Revision: 2.00
   SAL_B Revision: 2.13

   EFI Spec Revision: 1.10
   EFI Intel Drop Revision: 14.61
   EFI Build Revision: 2.10

   POSSE Revision: 0.10

   ACPI Revision: 7.00

   BMC Revision: 2.35                      ❷
   IPMI Revision: 1.00
   SMBIOS Revision: 2.3.2a
   Management Processor Revision: E.02.29  ❸
```

❶ The system firmware revision is 2.13.

❷ The BMC firmware revision is 2.35.

❸ The MP firmware revision is E.02.29.

The HP Integrity rx4640 server contains Dual Hot Plug Controller (DHPC) hardware with upgradable firmware. To check the current version of your DHPC firmware, use the EFI command INFO CHIPREV, as shown in the following example. The hot-plug controller version will be displayed. A display of 0100 indicates version 1.0; a display of 0110 means version 1.1.

```
Shell> INFO CHIPREV

CHIP REVISION INFORMATION

   Chip                  Logical     Device       Chip
   Type                     ID         ID       Revision
   -------------------   -------     ------     --------
   Memory Controller         0       122b         0023
   Root Bridge               0       1229         0023
     Host Bridge          0000       122e         0032
     Host Bridge          0001       122e         0032
     Host Bridge          0002       122e         0032
     Host Bridge          0004       122e         0032
      HotPlug Controller     0          0         0110
     Host Bridge          0005       122e         0032
      HotPlug Controller     0          0         0110
     Host Bridge          0006       122e         0032
     Other Bridge              0          0         0002
       Other Bridge            0          0         0008
         Baseboard MC          0          0         0235
```

For instructions on upgrading your firmware, refer to the *HP OpenVMS Version 8.2 Upgrade and Installation Manual*.

1.8.4 Booting On a Fibre Channel Storage Device

V8.2

Many customers prefer to boot from a Fibre Channel (FC) storage device because of its speed and because it can serve as a common cluster system disk in a SAN. Booting on a FC storage device on OpenVMS I64 systems is significantly different from booting on a FC storage device on OpenVMS Alpha systems.

Refer to the Fibre Channel chapter of *Guidelines for OpenVMS Cluster Configurations* for directions on how to configure and boot from a FC device on OpenVMS I64 systems.

1.8.5 Booting from the Installation DVD

V8.2

On I64 systems with the minimum amount of supported memory (512MB), the following message appears when booting from the installation DVD:

```
********* XFC-W-MemmgtInit Misconfigure Detected ********
XFC-E-MemMisconfigure MPW_HILIM + FREEGOAL > Physical Memory and no reserved memory for XFC
XFC-I-RECONFIG Setting MPW$GL_HILIM to no more than 25% of physical memory XFC-I-RECONFIG
Setting FREEGOAL to no more than 10% of physical memory
********* XFC-W-MemMisconfigure AUTOGEN should be run to correct configuration ********
********* XFC-I-MemmgtInit Bootstrap continuing ********
```

The message means that the system cache (XFC) initialization has successfully adjusted the SYSGEN parameters MPW_HILIM and FREEGOAL to allow caching to be effective during the installation. You can continue with the installation.

1.8.6 Setting Up I64 Systems to Reboot

V8.2

An I64 system does not automatically reboot unless it has been set up to do so from the console. In a cluster, the Alpha systems will reboot, but the I64 systems will shut down and sit at the console prompt.

For details about how to set up your I64 system to reboot, refer to the *HP OpenVMS Version 8.2 Upgrade and Installation Manual*.

1.8.7 HP DECwindows Motif Release Notes

The following DECwindows Motif release notes are of interest to OpenVMS I64 users.

1.8.7.1 Keyboard Support

V8.2

The only model keyboard supported on HP DECwindows Motif for OpenVMS I64 systems is the LK463 keyboard. Although other types of keyboards may function in the OpenVMS I64 environment, HP does not currently support them.

1.8.7.2 Connect Peripheral Devices Prior to Server Startup

V8.2

To properly configure your system as a DECwindows X display server, you must have all the following peripheral components connected prior to startup:

- monitor
- USB mouse
- USB keyboard

Otherwise, the server system might not complete the device initialization process correctly. For example, starting up a server system without input devices (mouse and keyboard) could result in a blank screen.

To correct this problem, disconnect and reconnect all peripherals.

1.8.7.3 Countdown Messages Displayed During Startup

V8.2

When running DECwindows Motif in client-only mode (with no server configured), messages similar to the following might be displayed during startup:

```
Waiting for mouse...
Waiting for keyboard...
```

These messages indicate that device polling is underway and are informational only. They will disappear once the 15-second countdown is complete.

To prevent the messages from displaying, connect the input devices (USB mouse and USB keyboard) to the system prior to startup.

1.9 Release Notes for OpenVMS Alpha Users

The following notes are primarily of interest to users of OpenVMS Alpha systems.

1.9.1 Firmware for OpenVMS Alpha Version 8.2

V8.2

OpenVMS Alpha Version 8.2 was tested with the platform-specific firmware included on Alpha Systems Firmware CD-ROM Version 6.8. For older platforms no longer included on the Firmware CD-ROM, OpenVMS Alpha Version 8.2 was tested with the latest released firmware version. HP recommends upgrading to the latest firmware before upgrading OpenVMS.

The OpenVMS Alpha Version 8.2 kit includes the Alpha Systems Firmware CD-ROM and Release Notes. Read the Release Notes before installing the firmware.

You can obtain Version 6.8 and the latest firmware information from the following website (URL is case sensitive):

`ftp://ftp.digital.com/pub/Digital/Alpha/firmware/`

1.9.2 Upgrade Paths

V8.2

You can upgrade *directly* to OpenVMS Alpha Version 8.2 from only the following versions of OpenVMS Alpha. Upgrades to OpenVMS I64 Version 8.2 are not supported.

Version 7.3-2
Version 7.3-1

If you are currently running OpenVMS Alpha Version 6.2*x* through 7.3, inclusive, you must first upgrade to Version 7.3-1 or 7.3-2, and then to Version 8.2. Note that standard support for OpenVMS Alpha Version 7.3-1 systems ends on March 31, 2005. After that, OpenVMS Alpha V7.3-1 systems will not be under Prior Version Support (PVS). For details about OpenVMS operating system support, see the chart on the following website:

`http://h71000.www7.hp.com/openvms/openvms_supportchart.html`

If you are running other versions of OpenVMS that are no longer supported, you must do multiple upgrades in accordance with upgrade paths that were documented for earlier versions.

Cluster Concurrent Upgrades

During a concurrent upgrade, you must shut down the entire cluster and upgrade each system disk. No one can use the cluster until you upgrade and reboot each computer. Once you reboot, each computer will be running the upgraded version of the operating system.

Cluster Rolling Upgrades

During a cluster rolling upgrade, you upgrade each system disk individually, allowing old and new versions of the operating system to run together in the same cluster (a mixed-version cluster). There must be more than one system disk. The systems that are not being upgraded remain available.

Only the following OpenVMS Alpha and OpenVMS VAX versions are supported in mixed-version clusters that include OpenVMS Alpha Version 8.2:

Version 7.3-2 (Alpha)
Version 7.3 (VAX)

If you are upgrading in a cluster environment, rolling upgrades are supported from Version 7.3-2 and 7.3-1 of the OpenVMS Alpha operating system. If you have other versions in a cluster, you cannot do a rolling upgrade until those versions are upgraded to a supported version.

Mixed-version support for all of these versions requires the installation of one or more remedial kits. For more information, see Section 4.18.4.

Note

HP currently supports only two versions of OpenVMS (regardless of architecture) running in a cluster at the same time. Only two architectures are supported in the same OpenVMS Cluster. Warranted support is provided for pairings with OpenVMS I64 Version 8.2. For more information, refer to the *HP OpenVMS Version 8.2 Upgrade and Installation Manual*.

For a discussion of warranted pairs and migration pairs of OpenVMS operating systems, for complete instructions for installing or upgrading to OpenVMS Alpha Version 8.2, and for instructions on installing OpenVMS I64 Version 8.2, refer to the *HP OpenVMS Version 8.2 Upgrade and Installation Manual*.

1.9.3 Error on Upgrade from Version 7.3-1

V8.2

When you upgrade an OpenVMS Alpha Version 7.3-1 system disk to Alpha Version 8.2, the following error message might appear during the execution phase:

```
%PCSI-E-PARUDF, file [SYSLIB]SYS$STARLET_C.TLB was not previously installed
or is present but out of scope; module update skipped
Terminating is strongly recommended.  Do you want to terminate? [YES]
```

This error message is not appropriate for the circumstances. It incorrectly states that the SYS$STARLET_C.TLB library cannot be updated. The cause of this problem is that a patch kit applied to the Alpha Version 7.3-1 system disk improperly updated a module in the library.

Disregard the recommendation to terminate the upgrade. Answer NO to the prompt `Do you want to terminate? [YES]`. The upgrade will continue successfully.

1.9.4 Encryption for OpenVMS Alpha: Remove Before Upgrading

V8.2

The Encryption producer name has changed from CPQ to HP for OpenVMS Alpha and Itanium systems. This name is visible in the PCSI filename and the PCSI database. For example, the following pairs demonstrate the old and new names:

CPQ-AXPVMS-ENCRYPT-V0106–1.PCSI
HP-AXPVMS-ENCRYPT-V0106–1.PCSI

CPQ AXPVMS ENCRYPT V1.6
HP AXPVMS ENCRYPT V1.6

For Alpha systems, if the ENCRYPT product was installed from an older kit (that is, a kit where HP is not listed as the Encryption producer), you must first remove ENCRYPT before you upgrade. Use the following commands:

```
$ PRODUCT REMOVE ENCRYPT
$ PRODUCT INSTALL ENCRYPT
```

A license is no longer required to use the the HP Encryption product.

1.9.5 HP DECram V3.*n*: Remove Before Upgrading

V8.2

Starting with OpenVMS Alpha and OpenVMS I64 Version 8.2, DECram ships with the OpenVMS operating system as a System Integrated Product (SIP). Before you upgrade to Version 8.2 on an OpenVMS Alpha system, you must remove any old versions of DECram. Refer to the *HP OpenVMS Version 8.2 Upgrade and Installation Manual* for details.

More DECram release notes are included in Section 2.13.

1.9.6 Kerberos V1.0: Remove Before Upgrading

V7.3-2

If you installed Kerberos Version 1.0 for OpenVMS using a POLYCENTER Software Installation kit, you must use the POLYCENTER Software Installation utility to remove Kerberos Version 1.0 before you upgrade the operating system. (You do *not* need to remove Kerberos if you are running Version 2.0 or if Version 1.0 was installed as part of the OpenVMS Version 7.3-1 operating system.)

To remove Kerberos, choose Option 6 ''Remove installed products'' from the installation CD main menu. During the removal, you are asked whether you want to remove the data and directories. (Data refers to the configuration data files along with the principal database, if one was created.) If you want to save this information for use later, respond ''No'' to the question. Return to the main menu and perform the upgrade of OpenVMS.

After the upgrade, the new Kerberos directories are located in KRB$ROOT:[*...]. (KRB$ROOT is defined as a system-wide logical name when Kerberos is started.) Kerberos data is either created during configuration or moved from the old Kerberos directories and renamed. If you removed a previously installed Kerberos kit and saved the data and directories, the data will automatically be moved into the new directories and be renamed the first time the Kerberos startup procedure is run after the upgrade.

Start the Kerberos servers by entering the following command:

```
$ @sys$startup:krb$startup.com
```

Note that the Kerberos startup procedure moves and renames only known Kerberos files. Users who have created files in the old Kerberos directories must manually move those files.

For more information about installing and configuring Kerberos, refer to *HP Open Source Security for OpenVMS, Volume 3: Kerberos*.

1.9.7 Converting the LANCP Device Database After Upgrading

V8.2

When you upgrade to OpenVMS Alpha Version 8.2, you might also need to convert the LAN device database to the Version 8.2 format if this is not automatically done by LANACP when LANACP is first run after the upgrade.

To convert the database, issue the following LANCP commands to convert the device database and then to stop LANACP so it can be restarted to use the new database:

```
$ LANCP
LANCP> CONVERT DEVICE_DATABASE
LANCP> SET ACP/STOP
LANCP> EXIT
$ @SYS$STARTUP:LAN$STARTUP
```

1.9.8 DECnet-Plus Requires a New Version

V7.3-2

When you install or upgrade to OpenVMS Alpha Version 7.3-2 or later, you must also install a new version of DECnet-Plus. One of the reasons that makes this necessary is a change of behavior in AUTOGEN (see Section 4.3).

Unlike the behavior of previous versions, DECnet-Plus for OpenVMS Version 7.3-2 and later now provides product information in NEWPARAMS.DAT records, as required by AUTOGEN. AUTOGEN anticipates this change in DECnet-Plus, so AUTOGEN does not print any warnings when it removes "bad" records from CLU$PARAMS.DAT; AUTOGEN presumes these records were made by an older DECnet-Plus kit and will be replaced by the new DECnet-Plus kit. So, under normal conditions, you will not see any striking differences in behavior during an OpenVMS Version 7.3-2 or later installation or upgrade.

However, if other products do not provide product information in NEWPARAMS.DAT records, as now required by AUTOGEN, AUTOGEN prints warning messages to both the report and the user's SYS$OUTPUT device. The warnings state that AUTOGEN cannot accept the parameter assignment found in NEWPARAMS.DAT (because no product name is attached) and that no records will be added to CLU$PARAMS.DAT. Since no records are added, the expected additions or other alterations to SYSGEN parameters will not be made, which could lead to resource exhaustion. Developers and testers of software products should be aware of this requirement; it may also be of interest to system managers.

This new behavior is intended to protect both the users and providers of layered products. By keeping this information ordered properly so that it can be updated properly, problems resulting from bad updates should be minimized.

A description of NEWPARAMS.DAT and CLU$PARAMS.DAT is included in the AUTOGEN chapter of the *HP OpenVMS System Management Utilities Reference Manual*.

1.9.9 SHADOW_MAX_UNIT Default Setting and Memory Usage

V7.3-2

This note updates an earlier note that discussed the default settings for this system parameter but did not describe the amount of main memory consumed by the default settings.

OpenVMS Alpha Version 7.3 introduced minicopy support in HP Volume Shadowing for OpenVMS. As part of the minicopy functionality, a new volume shadowing system parameter, SHADOW_MAX_UNIT, was introduced. On OpenVMS Alpha systems, the default value for this system parameter is 500, which consumes 24 KB of main memory. On OpenVMS VAX systems, the default value is 100, which consumes 5 KB of main memory.

If you do not plan to use Volume Shadowing for OpenVMS, you can change the setting to its minimum of 10 (which consumes 480 bytes of main memory). By setting the default to its minimum, you free up 23.5 KB of main memory on an OpenVMS Alpha system and 4.5 KB of main memory on a VAX system.

Note

SHAD_MAX_UNIT is a static system parameter. In order for a new setting to take effect, you must reboot your system.

Recommendations for SHADOW_MAX_UNIT settings for volume shadowing are discussed in the *HP Volume Shadowing for OpenVMS* manual.

2
OpenVMS Associated Products Release Notes

This chapter contains information about OpenVMS associated products. Notes specifically related to installation or upgrade issues related to associated products are in Chapter 1.

For notes about using compilers, linkers, and run-time library routines, see Chapter 5.

2.1 Associated Product Support

The Software Public Rollout Reports for OpenVMS list the availability of HP software products shipping on the Software Products Library kits (CD–ROM consolidations) for OpenVMS Alpha and OpenVMS VAX and on the Layered Product Library kits (DVD consolidation) for OpenVMS I64.

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. The information is not provided in these release notes because of the changing nature of the information.

The Software Public Rollout Reports for OpenVMS are available from the following website:

`http://h71000.www7.hp.com/openvms/os/swroll/`

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 files:

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

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

2.2 BASIC: V1.5A Required to Create STARLET Library (Alpha Only)

V7.3-2

Because of a change in OpenVMS Version 7.3-2 and later, BASIC versions prior to V1.5A cannot create the BASIC$STARLET library file during installation.

Earlier versions of BASIC can install on OpenVMS Version 7.3-2 and later provided you do not request the option to build the STARLET library file. Also, previously installed BASIC compilers and previously created STARLET library files will continue to function after upgrading an older OpenVMS system to Version 7.3-2 and later.

It is only the BASIC$STARLET library file creation that does not work on OpenVMS Version 7.3-2 and later. The BASIC V1.5A kit contains an enhanced installation procedure that correctly builds the STARLET library file on OpenVMS Version 7.3-2 and later.

BASIC V1.5A is available on the latest consolidated layered product CD-ROM.

2.3 CMAP Files Added

V8.2

The following new CMAP files are provided in the OpenVMS Version 8.2 internationalization data kit.

DECKANJI2000
GB18030
ISO8859-1-EURO
UTF8-20
UTF8-30

2.4 COBOL: Changes in I/O Run-Time Diagnostics and RMS Special Registers

V7.3

Because of the addition of Extended File Support in OpenVMS Alpha Version 7.2, you may notice changes in the handling of I/O run-time diagnostics and RMS special registers on OpenVMS Alpha Version 7.2 and higher. In particular, a long file name that produced RMS$_FNM under versions of OpenVMS Alpha prior to Version 7.2 now produces RMS$_CRE on OpenVMS Alpha Version 7.2 and higher. You do not need to use the new ODS-5 support to see the RMS differences.

2.5 COM for HP OpenVMS (Alpha Only)

The following release notes pertain to COM for HP OpenVMS.

2.5.1 COM for OpenVMS Support

V8.2

COM Version 1.4 for OpenVMS is currently supported on OpenVMS Alpha Version 7.3-2 and 8.2. For the latest information about COM for OpenVMS, refer to the following website:

`http://h71000.www7.hp.com/openvms/products/dcom/`

2.5.2 Registry Access Error with Heavy Load of Applications

V7.3-2

You might get an ‘‘Error accessing registry database, contact system manager (0x000025fc)’’ message if you run a heavy load of COM for OpenVMS applications with the CTLPAGES value set to 256 or less. Set the CTLPAGES value to 512 to avoid this problem.

2.6 DECdfs Version 2.4 Required for OpenVMS Version 8.2

V8.2

DECdfs Version 2.4 is required for OpenVMS Version 8.2. If you try to use an older version of DECdfs, you will get an error message.

2.7 DECforms Web Connector Version 3.0 (Alpha Only)

V7.3-1

If you already have DECforms installed, perform the following tasks to enable DECforms Web Connector V3.0 to run on OpenVMS Version 7.3-1 and higher:

1. Remove or comment out the following line:

   ```
   $ @SYS$COMMON:[JAVA$122.COM]JAVA$122_SETUP.COM
   ```

   from these command procedures in the FORMS$INSTALL_AREA directory:

   - FORMS_SMGR_STARTUP.COM
   - FORMS_WEB$STARTUP.COM
   - FORMS_WEB_CONFIG.COM

2. Ensure that the Java™ environment is set up systemwide for *all* processes. HP recommends adding the Java environment setup to the system's SYLOGIN.COM file.
3. Ensure that the browser clients use the Sun Java Plugin Version 1.2.2, as stated in the SPD and the Administrative guide.

2.8 DEC PL/I: RTL Support for OpenVMS

V7.3

There is a known incompatibility between the PL/I RTL distributed with the OpenVMS operating system and the more recent PL/I RTL owned and distributed by Kednos Corporation. The older version shipped with the OpenVMS operating system may overwrite a newer version. The image in question is SYS$LIBRARY:DPLI$RTLSHR.EXE.

OpenVMS distributes the following version of the file, which can be identified by using the DCL command ANALYZE/IMAGE:

```
Image Identification Information

        image name: "DPLI$RTLSHR"
        image file identification: "V4.0-6"
```

If you execute an ANALYZE/IMAGE command before upgrading to OpenVMS Version 7.3 or higher and find a newer version of DPLI$RTLSHR.EXE, you can either copy it and restore it after the upgrade or reinstall the PL/I kit afterward.

Any questions about DEC PL/I and VAX PL/I should be directed to Kednos Corporation:

Phone: (831) 373-7003
Email: tom@kednos.com

See a related note in Section 5.30.

2.9 FMS Kits

V8.2

You can install either of the following FMS kits (or later kits) on both OpenVMS Alpha and OpenVMS I64 systems:

Full kit: HPFMS025

Run-time kit: HPFMSRT025

2.10 Graphical Configuration Manager (GCM) (Alpha Only)

The Graphical Configuration Manager (GCM) is included on the Layered Products CD-ROM that ships with the operating system. However, GCM is frequently updated. Check regularly for new versions of the software on the following web page:

`http://h71000.www7.hp.com/openvms/products/gcm/`

2.11 HP DCE RPC for OpenVMS

The following notes pertain to OpenVMS Version 8.2 of HP DCE RPC.

2.11.1 DCE RPC Supports FAILSafe IP

V8.2

DCE RPC has been enhanced to work in a FAILSafe IP environment.

2.11.2 Support for Tuning the Buffer Size of RPC Sockets

V8.2

DCE RPC now provides support for tuning the socket buffer size by means of the logical RPC_USE_DEFAULT_SOCKET_BUFFER_SIZE. Setting this logical allows the RPC runtime to use the system default socket buffer size values. Use the following command to define the logical systemwide:

```
$ DEFINE/SYSTEM RPC_USE_DEFAULT_SOCKET_BUFFER_SIZE 1
```

To restore the original RPC run-time behavior, you must deassign the RPC_USE_DEFAULT_SOCKET_BUFFER_SIZE logical.

2.11.3 RTI (Remote Task Invocation) RPC (I64 Only)

V8.2

Version 8.2 of RPC does not support RTI RPC on I64 systems. For details, refer to the *HP DCE for OpenVMS Alpha and OpenVMS I64 Release Notes* document that ships with the kit.

2.11.4 Microsoft Lan Manager RPC Not Tested (I64 Only)

V8.2

Authenticated RPC over NTLM (Microsoft Lan Manager Protocol) has not been tested on OpenVMS I64 because the infrastructure on which DCE RPC depends is not available on OpenVMS I64.

2.11.5 The utc_mulftime Factor Argument Type

V8.2

The input argument, factor, for the DTSS API routine `utc_mulftime` must be an IEEE_FLOAT type on I64 systems and a G_FLOAT type on Alpha systems. You can use either CVT$FTOF or CVT$CONVERT_FLOAT to convert the factor argument to the appropriate floating-point type before calling `utc_mulftime`.

2.11.6 Support for G_FLOAT and IEEE Floating-Point Types

V8.2

DCE RPC for OpenVMS now supports both G_FLOAT and IEEE floating-point types on I64 and Alpha platforms.

Use the floating-point types consistently in a single RPC application. Different RPC applications, each using different floating-point types, can be run on a single system.

I64 Systems

By default, DCE uses IEEE_FLOAT type on I64 systems. That is, DCE applications built for I64 systems would use IEEE_FLOAT floating-point types.

To use the G_FLOAT floating-point type in RPC applications developed with the C or C++ language:

1. Call the new API function rpc_set_local_float_drep(RPC_APPLICATION_FLOAT_TYPE, &status) before calling any RPC run-time functions. The constant RPC_APPLICATION_ FLOAT_TYPE is automatically defined to the floating point type specified on the compiler command line qualifier. For details, refer to the *Release Notes for OpenVMS DCE V3.2*.
2. Compile the RPC application programs using the compiler qualifier /FLOAT=G_FLOAT.
3. Use the appropriate IDL compile option while building the stubs for the following:

   C applications: -CC_CMD "CC/FLOAT=G_FLOAT"

   C++ applications: -CPP_CMD "CXX/FLOAT=G_FLOAT"
4. Link the RPC applications using the appropriate DCE options file for the following:

   C applications: DCE.OPT

   C++ applications: DCE_CXX.OPT

To use the IEEE_FLOAT floating-point type in RPC applications developed with the C or C++ language:

1. Compile the RPC application programs using the compiler qualifier /FLOAT=IEEE_FLOAT (default option)
2. Link the RPC applications with DCE.OPT or with DCE_CXX.OPT.

Alpha Systems

By default, the DCE applications built on Alpha systems use the G_FLOAT floating-point types.

To use the IEEE_FLOAT floating-point type in RPC applications developed with the C or C++ language:

1. Call the new API function rpc_set_local_float_drep(RPC_APPLICATION_FLOAT_TYPE, &status) before calling RPC run-time functions. The constant RPC_APPLICATION_FLOAT_TYPE is automatically defined to the floating-point type specified on the compiler command line qualifier. For details, refer to the *Release Notes for OpenVMS DCE V3.2*.
2. Compile the RPC application programs using the compiler qualifier /FLOAT=IEEE_FLOAT.
3. Use the appropriate IDL compile option while building the stubs for the following:

   C applications: -CC_CMD "CC/FLOAT=IEEE_FLOAT"

   C++ applications: -CPP_CMD "CXX/FLOAT=IEEE_FLOAT"
4. Link the RPC applications using the appropriate DCE options file for the following:

   C applications: DCE.OPT

   C++ applications: DCE_CXX.OPT

To use the G_FLOAT floating-point type in RPC applications developed with the C or C++ language:

1. Compile the RPC application programs using the C or C++ compiler qualifier /FLOAT=G_FLOAT (default option).
2. Link the RPC application with DCE.OPT or DCE_CXX.OPT.

2.12 HP DECnet-Plus for OpenVMS: X.25 Data Links Not Supported (I64 Only)

V8.2

The HP X.25 for OpenVMS Alpha software is in the process of being ported and is not yet supported. Therefore, HP DECnet-Plus for OpenVMS I64 Version 8.2 does not support X.25 data links.

2.13 HP DECram

This section contains release notes pertaining to DECram.

2.13.1 DECram Ships With OpenVMS Version 8.2

V8.2

Starting with OpenVMS Alpha and OpenVMS I64 Version 8.2, DECram ships with the OpenVMS operating system as a System Integrated Product (SIP). Users are still required to have a DECram license. The DECram driver is located in SYS$COMMON:[SYS$LDR]. Alpha users should remove any remaining SYS$MDDRIVER images in the system-specific directories ([SYSx.SYS$LDR]).

For details about removing old versions of DECram before you upgrade to Version 8.2, refer to the *HP OpenVMS Version 8.2 Upgrade and Installation Manual*.

If you try to load any old versions of DECram, you will get the following error message:

SYSTEM-W-SYSVERDIF, system version mismatch; please relink

No older versions of DECram are supported on OpenVMS Version 8.2.

DECram Version 2.5 will continue to be supported on VAX systems only.

2.13.2 Conflict with DECRYPT DCL Command

V8.2

The Encryption for OpenVMS Alpha layered product creates its own DCL command DECRYPT at installation time. DECRYPT then overwrites the default definition of DECR, which users might have been using to invoke DECram.

If both products are installed, you can access the DECram interface by defining a DCL foreign command symbol such as the following:

```
$ DECRAM == "$MDMANAGER"
```

2.13.3 Maximum Disk Size for DECram

V7.3-2

The DECram documentation states that the largest value you can specify for /SIZE and /CAPACITY for DECram is 4,294,967,295 (or %XFFFFFFFF), depending on your available free memory.

However, the largest value that OpenVMS supports is 2,147,483,647 for ODS-2 and ODS-5 volumes.

2.13.4 DECram Commands and Errors

V7.3-1

It is important to check for disk errors after issuing any DECram command because not all errors are returned to the user interface. Errors specific to a device are sent to the system error log. Type SHOW DEVICE MD at the DCL prompt to see if any device errors were generated as a result of a DECram command. You will need to use an error log analyzer tool to recover the error. Errors are logged in ASCII file format; you can search for errors with an MD-E-FAILURE prefix in the SYS$SYSROOT:[SYSERR]ERRLOG.SYS file.

2.13.5 DECram and Volume Shadowing

V7.3-1

Using Volume Shadowing for OpenVMS, DECram can shadow a DECram disk to a physical disk. However, be aware that in the current implementation of Volume Shadowing for OpenVMS, if the physical disk goes away, you will be writing to a volatile disk. A mechanism that will ‘‘freeze writes’’ if the physical disk goes away is planned for a future release of Volume Shadowing for OpenVMS.

2.14 HP DECwindows Motif for OpenVMS

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

OpenVMS Associated Products Release Notes
2.14 HP DECwindows Motif for OpenVMS

2.14.1 DECwindows Pause Screen Can Fail to Unlock (Alpha Only)

V8.2

In an external authentication environment, a user will not be able to unlock a DECwindows workstation when all of the following conditions are met:

- The user is flagged as EXTAUTH in the SYSUAF database.
- The user logged into the DECwindows workstation using a username and the /LOCAL_PASSWORD qualifier.
- The pause screen has been activated.

Under these circumstances, the workstation does not recognize a valid password and keeps prompting for one. To recover from this condition, log in from another source (for example, on the network or LAT) and restart the DECwindows server using @SYS$STARTUP:DECW$STARTUP RESTART.

Until this problem is corrected in a future release, HP recommends that externally authenticated users who must log in to a DECWindows workstation using the /LOCAL_PASSWORD qualifier disable the automated pause function and refrain from invoking the pause screen.

2.14.2 New Locales Added

V8.2

The following new locales, which are used by localized DECwindows Motif software, have been added to the OpenVMS Version 8.2 internationalization data kit.

iw_IL.utf-8 (Hebrew, Israel, UTF-8)
ko_KR.utf-8 (Korean, UTF-8)
zh_CN.utf-8 (Chinese, PRC, UTF-8)
zh_HK.utf-8 (Chinese, Hong Kong, UTF-8)
zh_TW.utf-8 (Chinese, Taiwan, UTF-8)

2.14.3 Available Language Variants

V8.2

The only language variants offered with DECwindows Motif for OpenVMS Version 1.5 are Japanese, Hanzi, and Hangul. Hebrew will be offered later.

If you require another language variant for DECwindows Motif, contact your HP support representative either to identify a prior version that offers the variant or to discuss options regarding software translation.

2.14.4 Support for LAT Transport Interface

V8.2

Support for the DECwindows Motif interface to the LAT transport, which was withdrawn with DECwindows Motif Version 1.3, has been restored and is available on the OpenVMS Alpha and OpenVMS I64 platforms. This support enables users to start LAT X sessions and communicate over low-capacity networks with systems running DECwindows Motif Version 1.3-1 or higher. It also allows client applications running on DECwindows Motif systems to use the LAT transport to connect to X terminal systems.

Note that the restored LAT interface included with the OpenVMS operating system can be used as a valid network transport for communication with the DECwindows Motif Version 1.3-1 (or greater) and OpenVMS Version 7.3-2 (or greater) display servers. However, use with any other communication protocols in the X11R6.6 environment is not supported. This includes communication by or with the following:

- Inter-Client Exchange (ICE) and Session Manager protocols
- Low-Bandwidth X (LBX) proxy servers
- Proxy manager applications
- Font servers

Additionally, HP does not support the use of token-based authentication protocols (such as MIT-MAGIC-COOKIE-1 or MIT-KERBEROS-5) with the restored LAT transport interface.

2.14.5 User-Written Transports Not Supported

V7.3-2

In DECwindows Motif Version 1.3 for OpenVMS Alpha, significant changes were made to the DECwindows Motif transport library to accommodate multithreading and the communication needs of the Inter-Client Exchange (ICE) protocol, Low-Bandwidth X (LBX) proxy server, and Input Method servers. As a result, HP has discontinued support for user-written network transports on systems running DECwindows Motif Version 1.3 or higher.

All existing transports (DECNET, TCPIP, LAT, and LOCAL) remain available and function as expected. However, HP no longer provides support for designing and implementing user-written transports based on the updated transport interface. The *VMS DECwindows Transport Manual* has been archived, and the new libraries are not publicly available.

If you have implemented a custom transport and want to migrate that transport to the DECwindows Motif Version 1.5 or greater environment, contact your HP support representative to develop a migration strategy.

2.15 HP Secure Web Server Version Support

V8.2

OpenVMS Alpha Version 7.3-2 and OpenVMS Version 8.2 (Alpha and I64) are the last releases on which the Secure Web Server (SWS) Version 1.3-* is supported. OpenVMS Alpha Version 7.3-2 is the last release on which SWS Version 2.0 is supported.

The functional replacement for SWS Version 1.3-* and SWS Version 2.0 is SWS Version 2.1, which is expected to be available later in 2005. All future new features and enhancements to SWS will be provided beginning with SWS Version 2.1, which is based on the Apache 2.0.* open source code base.

SWS Version 1.3-* and SWS Version 2.0 will be supported while OpenVMS Alpha Version 7.3-2 is in Prior Version Support (PVS) status, and SWS Version 1.3-* will be supported as long as OpenVMS Version 8.2 is supported. Support for these SWS versions will include remedial fixes and security fixes as deemed appropriate.

2.16 Pascal

Pascal V5.9 is required for OpenVMS I64 systems.

The following release notes pertain to HP Pascal on OpenVMS Alpha systems.

2.16.1 V5.8A (or Later) Required to Create STARLET Library (Alpha Only)

V7.3-2

Because of a change in OpenVMS Version 7.3-2, Pascal versions prior to V5.8A cannot create the STARLET library files during installation.

Earlier versions of Pascal can install on OpenVMS Version 7.3-2 or later if you answer "NO" to the option to create and install the STARLET library files. Also, previously installed Pascal compilers and previously created STARLET library files will continue to function after upgrading an older OpenVMS system to Version 7.3-2 or later.

It is only the STARLET library creation portion of the Pascal installation that does not work on OpenVMS Version 7.3-2 and later. The Pascal V5.8A kit contains an enhanced installation procedure to correctly build the STARLET library files on OpenVMS Version 7.3-2 and later.

Pascal V5.8A is available on the latest consolidated layered product CD-ROM.

2.16.2 Installing HP Pascal After an Upgrade (Alpha Only)

V7.3

This note applies to any version of HP Pascal and any version of the OpenVMS Alpha operating system.

After upgrading OpenVMS, you should reinstall HP Pascal to produce new versions of STARLET.PAS and other definition files to match the upgraded system.

If you do not reinstall HP Pascal after upgrading OpenVMS, the compiler on your system will still work correctly. However, STARLET.PAS and the other definition files will not contain any new or corrected definitions supplied by the OpenVMS upgrade.

2.17 WEBES and SEA Support on I64 Systems

V8.2

The WEBased Enterprise Services (WEBES), a kitted suite of tools that includes the System Event Analyzer (SEA) for OpenVMS, is available on Alpha systems, and will be supported on I64 systems when the WEBES V4.4 I64 kit ships in early 2005. Once this kit ships, the System Tools CD that is packaged with OpenVMS quarterly updates will also be updated to include WEBES V4.4. The latest version of WEBES can always be obtained from the WEBES homepage at:

`http://www.hp.com/services/webes`

WEBES Version 4.3.3 and higher is qualified and supported to run on OpenVMS Alpha Version 8.2 now.

3
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 the *HP OpenVMS Version 8.2 New Features and Documentation Overview*.

3.1 OpenVMS Freeware CD–ROMs

V8.2

Included in the OpenVMS Version 8.2 media kit are the OpenVMS Freeware Version 7.0 CD-ROMs. The Freeware CD-ROMs contain free software tools and utilities for creating applications and for using and managing OpenVMS systems.

To mount the Freeware CD-ROMs, insert one of the CD-ROM volumes into the CD-ROM drive and enter the following commands to mount and display the contents of the Freeware volume. For additional information about the freeware, refer to the FREEWARE_README.TXT files.

```
$ MOUNT/OVERRIDE=IDENTIFICATION ddcu:
$ TYPE ddcu:[FREEWARE]FREEWARE_README.TXT
```

In these MOUNT commands, the *ddcu*: specification represents the device name of the CD-ROM or DVD-ROM device on your OpenVMS system.

Once the appropriate device is mounted, you can access the kit directories directly using standard DCL commands such as DIRECTORY and COPY. Text files containing submission abstracts and other materials are available in the [FREEWARE] directory on each disk.

3.1.1 Freeware Menu Unavailable (I64 Only)

V8.2

The [FREEWARE]FREEWARE.COM Freeware menu system interface on the OpenVMS Freeware V7.0 distribution does not operate on OpenVMS I64 systems.

The menu system interface is expected to function on OpenVMS Alpha and on OpenVMS VAX systems.

You can directly access the contents of the Freeware V7.0 distribution media by using DCL commands such as DIRECTORY and COPY from an OpenVMS I64, OpenVMS Alpha, or OpenVMS VAX system. This is the preferred way to access the contents of the Freeware V7.0 distribution.

Information about submissions and the Freeware distribution is contained in the file [FREEWARE]FREEWARE_README.TXT. This file is on each volume of the Freeware V7.0 distribution, and you can view its contents by using the TYPE command or a text editor.

3.2 Online Help Changes

V8.2

In OpenVMS Version 8.2, topics that were previously covered under the help topic DCL_Tips (which was previously found under HELP SPECIFY) have now been made into individual help topics that are accessible at the top level of help. Look for these new help topics:

HELP Date_Time (includes time specifications)
HELP Expression
HELP Filespec
HELP Integer
HELP Privilege
HELP Protection
HELP String
HELP Symbol
HELP UIC

Help for SET SERVER and SHOW SERVER has also been simplified and made more easily accessible. The technical content has not changed — except for SET SERVER ACME /CONFIGURE — but there are now separate help topics for the following:

SET SERVER ACME_SERVER
SET SERVER REGISTRY_SERVER
SET SERVER SECURITY_SERVER
SHOW SERVER ACME_SERVER
SHOW SERVER REGISTRY_SERVER

3.3 DCL Commands

The following release notes pertain to DCL commands.

3.3.1 ANALYZE/ERROR_LOG Command Not Ported to OpenVMS I64

V8.2

The ANALYZE/ERROR_LOG command has not been ported to OpenVMS I64. There are no plans to port this functionality to OpenVMS I64 in the future.

Online help erroneously includes the ANALYZE/ERROR_LOG command.

3.3.2 CREATE/MAILBOX: Temporary Restriction

V8.2

CREATE/MAILBOX/TEMPORARY currently requires CMEXEC privilege. This restriction will be removed in a future release.

3.3.3 DIAGNOSE Command No Longer Supported

V8.2

The DIAGNOSE command is not supported on OpenVMS Version 8.2. For more information, see Section A.4 and Section A.6.

3.3.4 SHOW LICENSE /OE and /HIERARCHY Require SYSLCK (I64 Only)

V8.2

Currently, the SHOW LICENSE/OE and SHOW LICENSE/HIERARCHY commands require the SYSLCK privilege. This restriction will be removed in a future release.

If you do not have SYSLCK privilege and want to know whether a specific product in the operating environment database is licensed to run on the system, you can use the lexical function F$LICENSE, as shown in this example:

```
$ WRITE SYS$OUTPUT F$LICENSE("RMSJNL")
TRUE
$
```

3.4 DECmigrate Not On Open Source Tools CD-ROM

V8.2

The OpenVMS Migration Software for VAX to Alpha (DECmigrate) is not included on the Open Source Tools CD shipped with the OpenVMS Version 8.2 distribution media. The software kit was included on the media for OpenVMS Version 7.3-2, but testing has not been completed on OpenVMS Version 8.2. The software will continue to be available on the following website for earlier versions of OpenVMS:

`http://h71000.www7.hp.com/openvms/products/omsva/omsva.html`

An update will be posted when support for OpenVMS Version 8.2 becomes available.

3.5 iconv Fixes

V8.2

The following fixes have been made to `iconv`:

- Previously, if an output file had a version limit, the `iconv` CONVERT command would not convert the file. This problem has been corrected.
- Previously, `iconv` converters for DEC Hanyu could not convert DTSCS (Digital Taiwan Supplemental Character Set). Starting with OpenVMS Version 8.2, `iconv` converters DECHANYU_EUCTW and EUCTW_DECHANYU can convert to and from DTSCS characters.

3.6 HP Secure Web Browser

The following notes pertain to the HP Secure Web Browser.

3.6.1 Increased Memory Required

V7.3-1

If you have an OpenVMS workstation and you are using the HP Secure Web Browser (SWB), based on Mozilla, the minimum memory requirement is 128 MB; however, 256 MB is highly recommended for more robust performance.

3.6.2 Installation Error on ODS-2 Disk Volume (I64 Only)

V8.2

Installing the HP Secure Web Browser (CSWB) Version 1.4 for OpenVMS I64 on an ODS-2 disk volume fails with a PCSI error, as follows:

```
%PCSI-E-OPENIN, error opening
ODS2$DISK:[SYS0.SYSCOMMON.][CSWB.RES]SAMPLE^.UNIXPSFONTS.PROPERTIES;* as input
-RMS-E-FND, ACP file or directory lookup failed
-SYSTEM-W-BADFILEVER, bad file version number
%PCSI-E-OPFAILED, operation failed
```

You can continue with the installation by answering "NO" to the "Do you want to terminate?" prompt. The installation will continue successfully.

As an alternative, you can install the HP Secure Web Browser on an ODS-5 disk volume.

3.7 TECO Editor Not Yet Available on I64 Systems

V8.2

The TECO editor is not yet supported for OpenVMS I64. HP expects this editor to be available in a future release of OpenVMS I64.

4
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 *HP OpenVMS Version 8.2 New Features and Documentation Overview*.

4.1 Recovering from System Hangs or Crashes (I64 Only)

V8.2

If your system hangs and you want to force a crash, press Ctrl/P from the console. The method of forcing a crash dump varies depending on whether XDELTA is loaded.

If XDELTA is loaded, pressing Ctrl/P causes the system to enter XDELTA. The system displays the instruction pointer and the current instruction. You can force a crash from XDELTA by entering ;C, as shown in the following example:

```
$
Console Brk at 8068AD40
8068AD40!       add      r16 = r24, r16 ;;  (New IPL = 3)
;C
```

If XDELTA is not loaded, pressing Ctrl/P causes the system to respond with the prompt ''Crash? (Y/N)''. Entering Y causes the system to crash. Entering any other character has no effect on the system.

4.2 AUTHORIZE: New Quotas for the DEFAULT and SYSTEM Account

V8.2

The quotas associated with the DEFAULT and SYSTEM account have been updated. These updated quotas will be seen only on fresh installations of OpenVMS or on the creation of a new SYSUAF data file. Existing SYSUAF data files will not be updated.

The updates to the DEFAULT account are as follows:

Quota	Old Value	New Value
ASTLM	250	300
BYTLM	64,000	128,000
ENQLM	2,000	4,000
FILLM	100	128

Quota	Old Value	New Value
PGFLQUOTA	50,000	256,000
TQELM	10	100
WSDEFAULT	2000	4,096
WSQUOTA	4000	8,192

The updates to the SYSTEM account are the same as the DEFAULT account with the exception of the following two quotas:

Quota	Old Value	New Value
BYTLM	64,000	256,000
PGFLQUOTA	50,000	700,000

For upgraded systems with existing SYSUAF files, the system manager might want to update the DEFAULT and SYSTEM account quotas to these new values.

4.3 AUTOGEN: New Behavior with NEWPARAMS.DAT Files

V7.3-2

AUTOGEN no longer allows layered product kits to provide NEWPARAMS.DAT records that do not include a product name. The most commonly used products that previously did not adhere to this rule are DECwindows and DECnet-Plus. You must install new versions of both products when you install OpenVMS Alpha Version 7.3-2 or later. (See Section 1.9.8.)

AUTOGEN looks for files named SYS$SYSTEM:NEWPARAMS.DAT;*, which contain SYSGEN parameter modifications specifying the consumption of system resources by layered products. A software installation kit may provide a NEWPARAMS.DAT file instead of telling the system manager to modify MODPARAMS.DAT to accommodate the requirements of the software being installed. For more information, refer to the AUTOGEN chapter in the *HP OpenVMS System Management Utilities Reference Manual*.

4.4 Backup Utility: Behavior Change

V8.2

A restore of a physical backup no longer requires the output disk to have the same geometry (tracks, cylinders). The restore works as long as the output has the same or larger capacity.

BACKUP/IMAGE is supported for I64 system disks. The image of an I64 system disk can be saved and restored on either an Alpha or an I64 system.

4.5 DECdtm/XA with Oracle® 8*i* and 9*i* (Alpha Only)

V7.3-2

When you are using DECdtm/XA to coordinate transactions with the Oracle® 8*i*/9*i* XA Compliant Resource Manager (RM), do not use the dynamic registration XA switch (xaoswd). Version 9.0.1.0.0 of the Oracle shareable library that supports dynamic registration does not work. Always use the static registration XA switch (xaosw) to bind the Oracle RM to the DECdtm/XA Veneer.

The DECdtm/XA V2.1 Gateway now has clusterwide transaction recovery support. Transactions from applications that use a clusterwide DECdtm Gateway Domain Log can now be recovered from any single-node failure. Gateway servers running on the remaining cluster nodes can initiate the transaction recovery process on behalf of the failed node.

4.6 Device Unit Number Maximum Increased

V8.2

In the past, OpenVMS would never create more than 10,000 cloned device units, and unit numbers would wrap after 9999. This had become a limitation for some devices, such as mailboxes or TCP/IP sockets.

Starting with OpenVMS Version 7.3-2, OpenVMS will create up to 32,767 devices if the DEV$V_NNM bit is clear in UCB$L_DEVCHAR2 and if bit 2 is clear in the DEVICE_NAMING system parameter. This does not require any device driver change. However, programs and command procedures that are coded to assume a maximum device number of 9999 may need to be modified.

4.7 ECP Data Collector and Performance Analyzer V5.5 (Alpha Only)

V8.2

Version 5.5 is the recommended version of the Enterprise Capacity and Performance (ECP) Analyzer for OpenVMS Alpha Version 8.2. Version 5.5 is backward compatible with OpenVMS Version 6.2 and higher.

The Performance Data Collector (TDC) Version 2.1 replaces the ECP Collector on OpenVMS Version 8.2. ECP Analyzer can analyze collection files created by TDC Version 2.1 and later.

The ECP Analyzer is currently not supported on OpenVMS I64.

4.8 EDIT/FDL: Fixing Recommended Bucket Size

V7.3

Prior to OpenVMS Version 7.3, when running EDIT/FDL, the calculated bucket sizes were always rounded up to the closest disk-cluster boundary, with a maximum bucket size of 63. This could cause problems when the disk-cluster size was large, but the ‘‘natural’’ bucket size for the file was small, because the bucket size was rounded up to a much larger value than required. Larger bucket sizes increase record and bucket lock contention, and can seriously impact performance.

OpenVMS Version 7.3 or higher modifies the algorithms for calculating the recommended bucket size to suggest a more reasonable size when the disk cluster is large.

4.9 EFI$CP Utility: Use Not Recommended

V8.2

The OpenVMS EFI$CP utility is presently considered undocumented and unsupported. HP recommends against use of this utility in OpenVMS Version 8.2. Certain privileged operations within this utility could render OpenVMS I64 unbootable.

4.10 EFI Tools: VMS_SHOW DUMP_DEV Errors (I64 Only)

V8.2

When a DUMP_DEV list is set by using the OpenVMS I64 Boot Manager utility (BOOT_OPTIONS.COM), and you then use the EFI utility to display the DUMP_ DEV EFI NVRAM variables from the EFI shell, VMS_SHOW.EFI displays "Error: Unknown Device" for each DUMP_DEV entry. To display the DUMP_DEV list, use the OpenVMS I64 Boot Manager utility. This problem will be fixed in the next release.

4.11 Error Log Viewer (ELV) Utility: TRANSLATE/PAGE Command

V7.3-2

If a message is signaled while you are viewing a report using the /PAGE qualifier with the TRANSLATE command, the display might become corrupted. The workaround for this problem is to refresh the display using Ctrl/W.

If you press Ctrl/Z immediately after a message is signaled, the program abruptly terminates. The workaround for this problem is to scroll past the signaled message before pressing Ctrl/Z.

4.12 External Authentication

This section contains release notes pertaining to external authentication. External authentication is an optional feature introduced in OpenVMS Version 7.1 that enables OpenVMS systems to authenticate designated users with their external user IDs and passwords. For detailed information about using external authentication, refer to the *HP OpenVMS Guide to System Security*. Also see Section 2.14.1 for a release note related to external authentication.

4.12.1 I64 External Authentication Support

V8.2

The Advanced Server for OpenVMS V7.3A ECO4 (and later) product kit contains standalone external authentication software for I64 systems in an OpenVMS cluster.

If you want to enable NT LAN Manager external authentication on OpenVMS Cluster member nodes running I64, you must copy the I64 standalone external authentication images from an Alpha system on which the Advanced Server is installed to the I64 member node, and complete the setup as described in the Advanced Server kit release notes.

4.12.2 SET PASSWORD Behavior Within a DECterm Terminal Session

V7.2

A DECterm terminal session does not have access to the external user name used for login and must prompt for one during SET PASSWORD operations. The external user name defaults to the process's OpenVMS user name. If the default is not appropriate (that is, if the external user name and mapped OpenVMS user name are different), you must enter the correct external user name.

The following example shows a SET PASSWORD operation initiated by a user with the external user name JOHN_DOE. The mapped OpenVMS user name is JOHNDOE and is the default used by the SET PASSWORD operation. In this

case, the default is incorrect and the actual external user name was specified by the user.

```
$ set password
External user name not known; Specify one (Y/N)[Y]? Y
External user name [JOHNDOE]: JOHN_DOE
Old password:
New password:
Verification:
%SET-I-SNDEXTAUTH, Sending password request to external authenticator
%SET-I-TRYPWDSYNCH, Attempting password synchronization
$
```

4.12.3 No Password Expiration Notification on Workstations

V7.1

In the LAN Manager domain, a user cannot log in once a password expires.

PC users receive notification of impending external user password expiration and can change passwords before they expire. However, when a user logs in from an OpenVMS workstation using external authentication, the login process cannot determine whether the external password is about to expire. Therefore, sites that enforce password expiration and whose users do not primarily use PCs can choose not to use external authentication for workstation users.

4.13 INITIALIZE Command Changes

V8.2

The default behavior of the INITIALIZE command has changed as follows:

- In previous OpenVMS versions, the /ERASE qualifier would both set the ERASE_ON_DELETE volume characteristic and perform an initial data security erase (DSE) of the volume. Starting with Version 8.2, users can independently select one or both of these actions by specifying either or both of the following new keywords with /ERASE:

/ERASE=INIT	Performs a DSE erase.
/ERASE=DELETE	Sets the ERASE_ON_DELETE flag on the volume.

  The default behavior of /ERASE with no keywords remains the same as in previous releases. (That is, /ERASE is equivalent to /ERASE=(INIT,DELETE).)

- The default behavior of INITIALIZE when /CLUSTER_SIZE is not specified has changed. For both ODS-2 and ODS-5, when no specific size is set, the default cluster size is now a minimum of 16. (Previously it was 3.) Also, when a cluster size is computed, the value is rounded up to the next multiple of 16. There is no behavior change when a value is specified for /CLUSTER_SIZE.

For another change to the INITIALIZE command, see Section 4.17. For detailed information about the INITIALIZE command, see online help or the *HP OpenVMS DCL Dictionary*.

4.14 INSTALL Utility

V8.2

Previously, programs linked with /TRACEBACK could not be installed using the /RESIDENT qualifier. This restriction has been removed. INSTALL now accepts the /RESIDENT qualifier on programs linked with /TRACEBACK, provided the program is installed without any privileges.

4.15 Lock Manager

The following notes pertain to the OpenVMS Distributed Lock Manager.

4.15.1 Lock Value Block Extended

V8.2

On Alpha and I64 systems, the lock value block has been extended from 16 to 64 bytes. To learn how to take advantage of this increase, refer to the following resources:

- *HP OpenVMS System Services Reference Manual*: See descriptions of $DEQ, $ENQ, and $GETLKI.
- *HP OpenVMS Programming Concepts Manual*: See information about mixed-version operation and interoperability.
- *HP OpenVMS System Analysis Tools Manual*: See the description of the SHOW RESOURCES display.

While use of this feature by applications is optional, the size of the RSB$ data structure defining each lock resource is increased by 48 bytes even when the feature is not being used.

4.15.2 Fast Lock Remastering and PE1

V7.3

The Lock Manager has a feature called lock remastering. A lock remaster is the process of moving the lock mastership of a resource tree to another node in the cluster. The node that masters a lock tree can process local locking requests much faster because communication is not required with another node in the cluster. Having a lock tree reside on the node doing the most locking operations can improve overall system performance.

Prior to OpenVMS Version 7.3, lock remastering resulted in all nodes sending one message per local lock to the new master. For a very large lock tree, it could require a substantial amount of time to perform the lock remastering operation. During the operation, all application locking to the lock tree is stalled.

Starting with OpenVMS Version 7.3, sending lock data to the new master is done with very large transfers. This is a much more efficient process and results in moving a lock tree from 3 to 20 times faster.

Only nodes running Version 7.3 or higher can use large transfers for lock remastering. Remastering between OpenVMS Version 7.3 or higher nodes and prior version nodes still requires sending a single message per lock.

If you currently use the PE1 system parameter to limit the size of lock trees that can be remastered, HP recommends that you either try increasing the value to allow large lock trees to move or try setting the value to zero (0) to allow any size lock tree to move.

4.16 Logical Disk (LD) Utility: Problem Fixed

V7.3-2

In previous versions of OpenVMS, the Logical Disk (LD) utility could bypass the cache for the container file. If RMS was used to read or write to the container file, RMS would have stale data if the LD utility was used to connect to the file and subsequently to the logical disk being written.

This problem occurred mainly when the LD utility was used to create disk images to be burned onto a CD-ROM.

This problem has been fixed and you no longer need to use the following DCL command to turn caching off for any file that will be used as a container file for the LD utility:

```
$ SET FILE/CACHING_ATTRIBUTE=NO_CACHING CONTAINER_FILE.DSK.
```

4.17 On-Disk Structure (ODS) Layout Changes

V8.2

Starting with OpenVMS Version 8.2, the DCL command INITIALIZE has a new /GPT qualifier that is activated by default on I64 systems to create a new system file called GPT.SYS. This new GPT.SYS file contains the OpenVMS bootblock information. (GPT stands for global unique identifier partition table.)

On Alpha systems, /NOGPT is the default. For details about the INITIALIZE command, consult online help or the *HP OpenVMS DCL Dictionary*.

When GPT.SYS is present, the volume logical block layout of the volume is different from how it has been on pre-Version 8.2 Alpha systems. GPT.SYS maps the first and last logical blocks on the volume (that is, LBN 0 and LBN MAXBLOCK-1 [based on the value of the F$GETDVI item code MAXBLOCK]) into two segments with a minimum size of 34 blocks each. For example, on a volume with /CLUSTER_SIZE=8, the map area output might look as follows:

```
DUMP/HEADER/BLOCK=COUNT=0 [000000]GPT.SYS
     :
     .
     Map area
            Retrieval pointers
                     Count:      40    LBN:          0
                     Count:      40    LBN:   71132920
```

If you plan to run existing programs on OpenVMS I64 or on an Alpha system with a GPT.SYS file, you should modify the first block (VBN 1) maps in any programs that assume the first block of an [000000]INDEXF.SYS file maps to LBN 0; with GPT.SYS, VBN 1 might no longer map the boot block. You may not need to change programs that make no association between LBN 0 and VBN 1 of INDEXF.SYS and that always use the VBN layout of INDEXF.SYS, but you should examine them to be certain.

The VBN layout of INDEXF.SYS is currently documented in Section 1.2.1 of the *Guide to OpenVMS File Applications* and in Section 2.5.1 of *VMS File System Internals* by Kirby McCoy (ISBN 1-55558-056-4, 1990).

Note

If you specify /GPT, the disk will not mount on systems running versions older than OpenVMS Version 7.2.

For other changes to the INITIALIZE command, refer to Section 4.13. For full details about INITIALIZE, see online help or the *HP OpenVMS DCL Dictionary*.

4.18 OpenVMS Cluster Systems

The release notes in this section pertain to OpenVMS Cluster systems.

4.18.1 OpenVMS I64 Cluster Support

V8.2

With few exceptions, OpenVMS Cluster software provides the same features on OpenVMS I64 systems as it offers on OpenVMS Alpha and OpenVMS VAX systems.

4.18.2 Temporary Exceptions

V8.2

The following exceptions are temporary:

- The number of systems permitted in a cluster depends on the platform configuration, as follows:
  - Maximum of 8 I64 systems
  - Maximum of 16 Alpha and I64 systems in a mixed-architecture cluster (not to exceed 8 I64 systems)

  Support for more than eight I64 systems in a cluster will be announced during the first half of 2005 (up to the maximum of 96 total nodes).
- A supported production cluster containing an I64 system cannot include a VAX system. VAX systems can be included in these clusters for the purposes of development and migration with the understanding that any problems arising from the existence of VAX systems in these clusters will result in the need for either the VAX or I64 systems to be removed. See the OpenVMS Cluster Software SPD for more information.
- Currently, only two architectures are allowed for supported production environments in an OpenVMS Cluster system. Refer to the *HP OpenVMS Version 8.2 Upgrade and Installation Manual* for a list of supported cluster configurations.
- Satellite booting for OpenVMS I64 systems, not available with this release, will be supported in a future release.

4.18.3 Permanent Exceptions

V8.2

OpenVMS Cluster software supports three proprietary interconnects on Alpha systems that are not supported on OpenVMS I64 systems: DSSI (DIGITAL Storage Systems Interconnect), CI (cluster interconnect), and Memory Channel. Although DSSI and CI storage cannot be directly connected to OpenVMS I64 systems, data stored on CI and DSSI disks (connected to Alpha systems) can be served to OpenVMS I64 systems in the same cluster.

Multihost shared storage on a SCSI interconnect, commonly known as SCSI clusters, is not supported. (It is also not supported on OpenVMS Alpha systems for newer SCSI adapters.) However, multihost, shared storage on industry-standard Fibre Channel is supported.

Note

Locally attached storage, on both OpenVMS Alpha systems (CI, DSSI, FC, or SCSI storage) and OpenVMS I64 systems (Fibre Channel or SCSI storage), can be served to any other member of the cluster.

4.18.4 Patch Kits Needed for Cluster Compatibility

V8.2

Before introducing an OpenVMS Version 8.2 system into an existing OpenVMS Cluster system, you must apply certain patch kits (also known as remedial kits) to your systems running earlier versions of OpenVMS. If you are using Fibre Channel, XFC, or Volume Shadowing, additional patch kits are required. Note that these kits are version specific.

The versions listed in Table 4–1 are supported in either a warranted configuration or a migration pair configuration. For more information about these configurations, refer to either *HP OpenVMS Cluster Systems* or the *HP OpenVMS Version 8.2 Upgrade and Installation Manual*.

Table 4–1 lists the facilities that require patch kits and the patch ID names. Each patch kit has a corresponding readme file with the same name (file extension is .README).

You can either download the patch kits from the following web site (select the OpenVMS Software Patches option), or contact your HP support representative to receive the patch kits on media appropriate for your system:

`http://h18007.www1.hp.com/support/files/index.html`

Note

Patch kits are periodically updated on an as-needed basis. Always use the most recent patch kit for the facility, as indicated by the version number in the kit's readme file. The most recent version of each kit is the version posted on the web site.

Table 4–1 Patch Kits Required for Cluster Compatibility

Facility	Patch ID
OpenVMS Alpha Version 7.3-2	
Update kit with most patch kits except those also listed in this section	VMS732_UPDATE-V0300
Fibre Channel/SCSI	VMS732_FIBRE_SCSI-V0400

(continued on next page)

Table 4–1 (Cont.) Patch Kits Required for Cluster Compatibility

Facility	Patch ID
OpenVMS Alpha Version 7.3-2	
System	VMS732_SYS-V0600
OpenVMS VAX Version 7.3 [1]	
Audit Server	VAXAUDS01_073
Cluster	VAXSYSL01_073
DECnet-Plus	VAX_DNVOSIECO04-V73
DECwindows Motif	VAXDWMOTMUP01_073
DTS	VAXDTSS01_073
Files 11	VAXF11X02_073
MAIL	VAXMAIL01_073
MIME	VAXMIME01_073
MOUNT	VAXMOUN01_073
RMS	VAXRMS01_073
RPC	VAXRPC02_073
Volume Shadowing	VAXSHAD01_073
System	VAXSYS01_073

[1]For operating guidelines when using VAX systems in a cluster, refer to Section 4.18.2.

Note that VAX systems cannot be in a cluster with I64 systems. For a complete list of warranted groupings within a cluster, refer to the *HP OpenVMS Version 8.2 Upgrade and Installation Manual*.

4.18.5 New API Can Correct Incompatibility of Fibre Channel and SCSI Multipath with Some Third-Party Products

V7.3-2

OpenVMS Alpha Version 7.2-1 introduced the multipath feature, which provides support for failover between the multiple paths that can exist between a system and a SCSI or Fibre Channel device. OpenVMS Alpha Version 7.3-1 introduced support for failover between Fibre Channel multipath tape devices.

This multipath feature can be incompatible with some third-party disk-caching, disk-shadowing, or similar products. HP advises that you do not use such software on SCSI or Fibre Channel devices that are configured for multipath failover until this feature is supported by the producer of the software.

Third-party products that rely on altering the Driver Dispatch Table (DDT) of either the OpenVMS Alpha SCSI disk class driver (SYS$DKDRIVER.EXE), the OpenVMS Alpha SCSI tape class driver (SYS$MKDRIVER.EXE), or the SCSI generic class driver (SYS$GKDRIVER) may need to be modified in order to function correctly with the SCSI multipath feature.

Producers of such software can now modify their software using new DDT Intercept Establisher routines introduced in OpenVMS Alpha Version 7.3-2. For more information about these routines, refer to the *HP OpenVMS Alpha Version 7.3–2 New Features and Documentation Overview* manual.

Note

If you are using a third-party disk-caching product or disk shadowing application, refrain from using it in an OpenVMS SCSI or Fibre Channel multipath configuration until you confirm that the application has been revised using these new routines.

For more information about OpenVMS Alpha SCSI and Fibre Channel multipath features, refer to *Guidelines for OpenVMS Cluster Configurations*.

4.18.6 CLUSTER_CONFIG.COM and Limits on Root Directory Names

V7.3-2

This note updates Table 8-3 (Data Requested by CLUSTER_CONFIG_LAN.COM and CLUSTER_CONFIG.COM) in the *HP OpenVMS Cluster Systems* manual.

The documentation specifies a limit on the number of hexadecimal digits you can use for computers with direct access to the system disk. The limit is correct for VAX computers but not for Alpha computers.

The command procedure prompts for the following information:

```
Computer's root directory name on cluster system disk:
```

The documentation currently states:

Press Return to accept the procedure-supplied default, or specify a name in the form SYS*x*:

- For computers with direct access to the system disk, *x* is a hexadecimal digit in the range of 1 through 9 or A through D (for example, SYS1 OR SYSA)
- For satellites, *x* must be in the range of 10 through FFFF

The limit on the range of hexadecimal values with direct access to the system disk is correct for VAX computers. For Alpha computers with direct access to the system disk, the valid range of hexadecimal values is much larger. It includes both the VAX range of 1 through 9 or A through D, and also the range 10 through FFFF. Note that SYSE and SYSF are reserved for system use.

The *HP OpenVMS Cluster Systems* manual will include this information in its next revision.

4.18.7 Cluster Performance Reduced with CI-LAN Circuit Switching

V7.3-1

In rare cases, in an OpenVMS Cluster configuration with both CI and multiple FDDI, 100 Mb/s or Gb/s Ethernet-based circuits, you might observe that SCS connections are moving between CI and LAN circuits at intervals of approximately 1 minute. This frequent circuit switching can result in reduced cluster performance and may trigger mount verification of shadow set members.

PEdriver can detect and respond to LAN congestion that persists for a few seconds. When it detects a significant delay increase or packet losses on a LAN path, the PEdriver removes the path from use. When it detects that the path has improved, it begins using it again.

Under marginal conditions, the additional load on a LAN path resulting from its use for cluster traffic may cause its delay or packet losses to increase beyond acceptable limits. When the cluster load is removed, the path might appear to be sufficiently improved so that it will again come into use.

If a marginal LAN path's contribution to the LAN circuit's load class increases the circuit's load class above the CI's load class value of 140 when the marginal path is included (and, conversely, decreases the LAN circuit's load class below 140 when the path is excluded), SCS connections will move between CI and LAN circuits.

You can observe connections moving between LAN and CI circuits by using SHOW CLUSTER with the CONNECTION and CIRCUITS classes added.

Workarounds

If excessively frequent connection moves are observed, you can use one of the following workarounds:

- You can use SCACP or Availability Manager to assign a higher priority to the circuit, or the port you wish to be used, thus overriding automatic connection assignment and moving.

  Examples of SCACP commands are:

```
$ MC SCACP
SCACP> SET PORT PNA0 /PRIORITY=2    ! This will cause circuits from local
                                    ! CI port PNA0 to be chosen over
                                    ! lower priority circuits.

SCACP> SET PORT PEA0 /PRIORITY=2    ! This will cause LAN circuits to be
                                    ! chosen over lower priority circuits.
```

- You can use the SCACP SHOW CHANNEL commands to determine which channels are being switched into/out of use. Then you can use SCACP to explicitly exclude a specific channel by assigning it a lower priority value than the desired channels. For example:

```
SCACP> SET CHANNEL LARRY /LOCAL=EWB/REMOTE=EWB /PRIORITY=-2
```

  Note that CHANNEL and LAN device priority values in the range of *max, max-1* are considered equivalent; that is, they are treated as if they both had the maximum priority value. A difference of 2 or more in priority values is necessary to exclude a channel or LAN device from use.

4.18.8 Gigabit Ethernet Switch Restriction in an OpenVMS Cluster System

Permanent Restriction

Attempts to add a Gigabit Ethernet node to an OpenVMS Cluster system over a Gigabit Ethernet switch fail if the switch settings are different from the setting of the Gigabit Ethernet adapter with respect to autonegotiation. If the switch is set to autonegotiation, the adapter must be as well, and conversely.

Most Gigabit Ethernet adapters default to having autonegotiation enabled. An exception is the DEGXA on Alpha systems where the EGn0_MODE console environment variable contains the desired setting, which must match the switch setting.

When an attempt to add a node fails because of the switch and adapter mismatch, the messages that are displayed can be misleading. If you are using CLUSTER_CONFIG.COM to add the node and the option to install a local page and swap disk is selected, the problem might look like a disk-serving problem. The node running CLUSTER_CONFIG.COM displays the message "waiting for *node-name* to boot," while the booting node displays "waiting to tune system." The list of available disks is never displayed because of a missing network path. The network path is missing because of the autonegotiation mismatch between the Gigabit adapter and the switch.

To avoid this problem, disable autonegotiation on the new node's Gigabit Ethernet adapter, as follows:

1. Perform a conversational boot when you first boot the node into the cluster.
2. Set the new node's system parameter LAN_FLAGS to a value of 32 to disable autonegotiation on all Gigabit adapters in the system.

After this initial configuration, the LAN_FLAGS system parameter setting for autonegotiation must be consistent with the switch settings for all Gigabit Ethernet adapters in the system. If not all adapters need to disable autonegotiation, the run-time settings must be set appropriately in the LANCP device database. See the LANCP chapter in the *HP OpenVMS System Management Utilities Reference Manual* for details.

4.18.9 Multipath Tape Failover Restriction

V7.3-1

While the INITIALIZE command is in progress on a device in a Fibre Channel multipath tape set, multipath failover to another member of the set is not supported. If the current path fails while another multipath tape device is being initialized, retry the INITIALIZE command after the tape device fails over to a functioning path.

This restriction will be removed in a future release.

4.18.10 No Automatic Failover for SCSI Multipath Medium Changers

V7.3-1

Automatic path switching is not implemented in OpenVMS Alpha Version 7.3-1 or higher for SCSI medium changers (tape robots) attached to Fibre Channel using a Fibre-to-SCSI tape bridge. Multiple paths can be configured for such devices, but the only way to switch from one path to another is to use manual path switching with the SET DEVICE/SWITCH command.

This restriction will be removed in a future release.

4.19 OpenVMS Galaxy (Alpha Only)

The following sections contain notes pertaining to OpenVMS Galaxy systems.

Note that OpenVMS Galaxy is supported on OpenVMS Alpha systems only.

System Management Release Notes
4.19 OpenVMS Galaxy (Alpha Only)

4.19.1 Galaxy Definitions

V8.2

Because the *HP OpenVMS Alpha Partitioning and Galaxy Guide* is not being updated for OpenVMS Version 8.2, this note provides improved definitions of the word Galaxy, which depends on context.

Table 4–2 Galaxy Definitions

Galaxy as a:	Functions this way:
License	Is required to create and run multiple instances of OpenVMS in a single computer. Without this license, only one instance of OpenVMS can be run in a single computer.
System parameter	Sets memory sharing. GALAXY set to 1 specifies that OpenVMS instances with the parameter set in a hard partition will share memory between soft partitions within that hard partition. (You can run more than two soft partitions in a hard partition, and you may not want to share memory among all of them.) Note that this parameter only specifies whether a node uses shared memory. There is no need to use this parameter to run multiple, cooperative instances of OpenVMS; this is achieved by console setup of the desired configuration tree. GALAXY set to 0 means that memory is not shared (the default).
Soft partition	Provides the capability of several OpenVMS instances to execute cooperatively in a single computer so as to be able to migrate CPUs, use APIs, share memory, and so on. Platform partitioning makes possible the separation of resources into multiple soft partitions, each of which can run an OS instance. A soft partition is that subset of resources that the OS instance running in it can see and use.

4.19.2 OpenVMS Graphical Configuration Manager

V8.2

The OpenVMS Graphical Configuration Manager (GCM) is now supported for AlphaServer ES47/ES80/GS1280 Galaxy configurations. Previously, only the Graphical Configuration Utility (GCU) was supported.

4.19.3 Galaxy on ES40: Uncompressed Dump Limitation

Permanent Restriction

On AlphaServer ES40 Galaxy systems, you cannot write a raw (uncompressed) dump from instance 1 if instance 1's memory starts at or above 4 GB (physical). Instead, you must write a compressed dump.

4.19.4 Galaxy on ES40: Turning Off Fast Path

V7.3-1

When you implement Galaxy on an AlphaServer ES40 system, you must turn off Fast Path on instance 1. Do this by setting the SYSGEN parameter FAST_PATH to 0 on that instance.

If you do not turn off Fast Path on instance 1, I/O on instance 1 will hang when instance 0 is rebooted. This hang will continue until the PCI bus is reset and instance 1 rebooted. If there is shared SCSI or Fibre Channel, I/O will hang on the sharing nodes and all paths to those devices will be disabled.

4.20 OpenVMS Management Station

V8.2

Version 3.2D is the recommended version of OpenVMS Management Station for OpenVMS I64 Version 8.2 and OpenVMS Alpha Version 8.2. However, OpenVMS Management Station is backward compatible with OpenVMS Version 6.2 and higher.

The OpenVMS Version 8.2 installation includes OpenVMS Management Station Version 3.2D.

4.21 OpenVMS Registry Can Corrupt Version 2 Format Database

V7.3-2

If you create eight or more volatile subkeys in a key tree and then reboot a standalone system or a cluster, the OpenVMS Registry server can corrupt a Version 2 format Registry database when the server starts up after the reboot.

To avoid this problem, do one of the following:

- Do not use volatile keys.
- Use a Version 1 format database.

Note that Advanced Server for OpenVMS and COM for OpenVMS do not create volatile keys.

4.22 Security: Changes to DIRECTORY Command Output

V7.3-2

In OpenVMS Version 7.1 and higher, if you execute the DCL command DIRECTORY/SECURITY or DIRECTORY/FULL for files that contain Advanced Server (PATHWORKS) access control entries (ACEs), the hexadecimal representation for each Advanced Server ACE is no longer displayed. Instead, the total number of Advanced Server ACEs encountered for each file is summarized in the message, ''Suppressed *n* PATHWORKS ACEs.''

To display the suppressed ACEs, use the SHOW SECURITY command. You must have the SECURITY privilege to display these ACEs. Note that, in actuality, the command displays OpenVMS ACEs, including the %x86 ACE that reveals the Windows NT® security descriptor information. The Windows NT security descriptor information pertains to the Advanced Server.

4.23 Server Management Process (SMHANDLER)

V7.3-2

The server management process, SMHANDLER, now starts automatically on Alpha systems that support it. System managers should remove references to the obsolete startup file, SYS$STARTUP:SYS$SMHANDLER_STARTUP.COM, from SYSTARTUP_VMS.COM or other site-specific startup files. This reference has been removed from SYSTARTUP_VMS.TEMPLATE.

Background: What is SMHANDLER?

On certain Alpha systems, the server management process is started to assist the system firmware in reporting and responding to imminent hardware failures. Failure conditions vary but typically include over-temperature conditions, fan failures, or power supply failures. SMHANDLER may report warning conditions to OPCOM, and may initiate a shutdown of OpenVMS if system firmware is about to power off a failing system. In most situations, a controlled shutdown of OpenVMS would be less disruptive than abrupt loss of system power.

To ensure the longest possible up time, system managers can set the POWEROFF system parameter to 0. This prevents SMHANDLER from shutting down OpenVMS on a failing system but does not prevent system firmware from powering off the system.

4.24 SYSGEN: Security Auditing Fixed

V7.3-2

Previously, enabling SYSGEN audits, or alarms, did not provide any audits or alarms with information about the parameters being modified. As of OpenVMS Version 7.3-2, this problem is corrected. Audits or alarms now provide a list of the changed parameters along with their old and new values.

4.25 SYSMAN Utility: DUMP_PRIORITY Command Changes

V8.2

The following changes have been made to the SYSMAN DUMP_PRIORITY command:

- DUMP_PRIORITY ADD, MODIFY, and REMOVE

  The new qualifier /[NO]INFORMATIONAL has been added to allow users to control the output of informational messages, for example, in command procedures. The default is /INFORMATIONAL.

- DUMP_PRIORITY ADD and MODIFY

  If you attempt to add an entry that already exists, or if you use DUMP_ PRIORITY MODIFY to modify an existing entry where the modification would result in a duplicate, you will get an "SMI-I-SDPRDUPIGN, duplicate record creation ignored" message. (Previously, you would have gotten an "SMI-F-SDPRNODUP, duplicate records not allowed" message.) In the case of MODIFY, the existing record will not be removed.

- DUMP_PRIORITY REMOVE

  If you try to remove a nonexistent entry from the System Dump Priority registry, you now get an "SMI-I-SDPRNOTREM, no record removed" message. (Previously, you would have gotten an "SMI-F-SDPRNOTFOUND, system dump priority record not found" message, which is still returned by DUMP_ PRIORITY MODIFY when the entry to modify is not found.)

4.26 System Parameters

The following sections contain notes related to system parameters.

4.26.1 New System Parameters

V8.2

The following system parameters are new in OpenVMS Version 8.2:

ERLBUFFERPAG_S2
ERRORLOGBUFF_S2
SCSI_ERROR_POLL
SHADOW_HBMM_RTC
SHADOW_REC_DLY
SHADOW_SITE_ID
SYSSER_LOGGING
TTY_DEFCHAR3
VHPT_SIZE (I64 only)

The DEVICE_NAMING parameter was introduced in an earlier release, but was inadvertently omitted from the documentation; it is now included.

Refer to online help or the *HP OpenVMS System Management Utilities Reference Manual* for definitions of all system parameters.

4.26.2 Obsolete System Parameters

V8.2

The following system parameters have been marked as obsolete in OpenVMS Version 8.2:

BJOBLIM
BOOT_STYLE
EXUSRSTK
LAMAPREGS
NISCS_LAN_OVRHD
NJOBLIM
QBUS_MULT_INTR
SA_APP
SBIERRENABLE
SCSCONNECT
SD_ALLOCLASS
TAILORED
UDABURSTRATE
VECTOR_MARGIN
VECTOR_PROC
XFMAXRATE

4.26.3 Displaying Obsolete System Parameters

V8.2

SHOW commands in the SYSGEN, SYSMAN, SDA, and SYSBOOT utilities no longer display obsolete system parameters unless you do one of the following:

- Include the new /OBSOLETE qualifier on the SHOW command. For example, to display all obsolete system parameters, enter the following command:

```
SYSGEN> SHOW/OBSOLETE
```

- Specify the exact name of an obsolete parameter in a SHOW *parameter-name* command. For example, the following command displays OBSOLETE if the specified parameter is obsolete:

  ```
  SYSGEN> SHOW parameter-name
  ```

4.26.4 System Parameter Changes

V8.2

In OpenVMS Version 8.2, a number of system parameters are updated to provide optimum default, minimum, and maximum values.

- Support for 32,767 processes

  OpenVMS now supports a maximum of 32,767 processes. The maximum value for the following system parameters is increased to 32765 to support this change:

  BALSETCNT
  IJOBLIM
  MAXPROCESSCNT

- Nonpaged pool lookaside list trimming now turned off

  Trimming of the nonpaged pool lookaside lists is now turned off by default. Trimming of these lists can result in a fragmented nonpaged pool variable list. In some cases, a heavily fragmented variable pool list might result in degraded system performance. The default value for the following system parameters is now 100:

  NPAG_AGGRESSIVE
  NPAG_GENTLE

- Increase in minimum value of CTLPAGES

  The minimum value of CTLPAGES has been increased to 64, a value that can support a minimum boot.

- Larger CLISYMTBL values

  Both the minimum and maximum values of CLISYMTBL are increased (64 and 8192, respectively) to support the creation of additional and large DCL symbols. The new minimum value for CLISYMTBL also supports a minimum boot.

- Larger default values of CHANNELCNT

  The default and minimum values of CHANNELCNT increased to reduce the failure of applications that have a large number of files open because of an insufficient CHANNELCNT value. The new minimum is 64 and the new default is 512.

- Parameters associated with global sections increased

  System quotas associated with global sections have been increased to provide more appropriate default values:

  GBLPAGFIL (Default: 16394)
  GBLSECTIONS (Default: 1024)

- Default value of DUMPSTYLE changed for a compressed/selective dump

The default value of the DUMPSTYLE parameter is changed to 9 to produce a compressed/selective dump. This change helps ensure that necessary data is recorded in the system dump file when a system has a small system dump file.

- Default and minimum values of QUANTUM decreased

  With faster systems, the amount of work a CPU can do in a given amount of time has increased. Lowering the default value of QUANTUM to 5 allows a CPU to schedule more processes per second. The minimum value is 1.

- Values of the GH_* parameters increased

  The maximum value of the GH_* parameters is increased to 65536 (512MB). Note that it is impossible to boot a system with all of these parameters increased to the MAX value. These parameters consume a 32-bit address space, which is limited to 2 GB.

  I64 images have a read-only segment that can be shared. To support this, the GH_RES_DATA parameter default has been increased from 0 to 100 pages.

- Default value of WSMAX increased

  The default value of WSMAX is increased to 131072 to support the large physical memory on I64 systems.

- PQL parameter values increased on OpenVMS I64

  The PQL parameters are used to provide default and minimum values for various quotas when detached processes are created. A number of these parameters are increased to provide more reasonable defaults and minimums to accommodate the large physical memory on I64 systems. The new defaults are as follows:

Parameter	Default on Alpha	Default on I64
PQL_DBYTLM	65536	262144
PQL_MBYTLM	2048	128000
PQL_DPGFLQUOTA	131072	700000
PQL_MPGFLQUOTA	4096	512000
PQL_DWSDEFAULT	2048	32767
PQL_MWSDEFAULT	1024	16384
PQL_DWSQUOTA	4096	65536
PQL_MWSQUOTA	2048	32768
PQL_DWSEXTENT	32767	131072
PQL_MWSEXTENT	4094	65536
PQL_DENQLM	2048	2048
PQL_MENQLM	64	64

- Defaults for paged and nonpaged pool increased for I64

The default values for a number of the paged and nonpaged pool parameters are increased for I64, as follows:

Parameter	Default
NPAGEDYN	4194304 (4 MB)
NPAGEVIR	16777216 (16 MB)
PAGEDYN	4194304 (4 MB)

- Default of the system working set quota increased for I64

  The default value for SYSMWCNT is increased to 8192.

- Default for KSTACKPAGES increased for Alpha

  The default value of KSTACKPAGES is increased from 1 page to 2. This change is to ensure that various hardware platforms with assorted devices can boot with the default value.

Refer to online help or *HP OpenVMS System Management Utilities Reference Manual* for detailed descriptions of these parameters.

4.26.5 MMG_CTLFLAGS: Documentation Errors

V8.2

The description of bit 0 of system parameter MMG_CTLFLAGS is incorrect in online help, the *HP OpenVMS System Management Utilities Reference Manual*, and the *OpenVMS Performance Management* manual. The correct description of bit 0 is as follows:

> "If this bit is set, reclamation is enabled by trimming from periodically executing, but otherwise idle, processes. This occurs when the size of the free list plus the modified list drops below two times the value of FREEGOAL. This function is disabled if the bit is clear."

There is also an error in the description of Bit 1 in the *OpenVMS Performance Management* manual. That description should be corrected to read as follows:

> "Reclamation enabled by outswapping processes that have been idle for longer than LONGWAIT seconds. This occurs when the size of the free list drops below the value of FREEGOAL."

4.27 Terminal Fallback Facility (TFF)

On OpenVMS Alpha systems, 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, enter 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$FBDRIVER.EXE. On VAX systems, the TFF fallback driver is named FBDRIVER.EXE.
- On Alpha systems, TFF is capable of handling 16-bit character fallback. The OpenVMS Alpha fallback table library (TFF$MASTER.DAT) contains four more 16-bit character tables than the VAX library. Table 4–3 describes these additional tables.

Table 4–3 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.

- 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 now archived *OpenVMS Terminal Fallback Utility Manual* on the OpenVMS documentation website:

`http://www.hp.com/go/openvms/doc`

Click on ‘‘Archived documents’’ in the left sidebar to link to this manual.

4.28 Time Zone Changes

V8.2

In OpenVMS Version 8.2, 204 time zones have been added and existing time zones have been updated for a total of 540 time zones. This list of time zones is based on the time zone public database tzdata2003e.tar.gz, which is available from:

`ftp://elsie.nci.nih.gov/pub/`

An appendix listing all the time zones is included in *HP OpenVMS System Manager's Manual, Volume 2: Tuning, Monitoring, and Complex Systems*.

4.29 User Environment Test Package (UETP) (I64 Only)

V8.2

The User Environment Test Package (UETP) can be used with the following cautions:

- During the load phase, there are sporadic access violations in UETMEMY01. This does not terminate execution or really affect the validity of the run. UETP is still useable and produces valid results.
- The device phase currently does not complete execution due to an access violation.
- The DECnet phase runs fine. The cluster phase is still being tested. It appears to execute properly, but there are some concerns, and the output does not show other system names properly.

4.30 Recommended Caching Methods

V8.2

Virtual I/O Cache (VIOC) — also known as VAX Cluster Cache (VCC) — is not available on OpenVMS I64. On I64 systems, setting the SYSGEN parameter VCC_FLAGS to 1 is equivalent to setting VCC_FLAGS to 0 or not loading caching at all.

HP recommends Extended File Cache (XFC) as the preferred method for caching on both Alpha and I64 systems. For more information about XFC, refer to the *HP OpenVMS System Manager's Manual*.

In a future release of OpenVMS Alpha, support for VIOC will be removed.

4.31 Volume Shadowing for OpenVMS

The following release notes pertain to HP Volume Shadowing for OpenVMS, also known as host-based volume shadowing (HBVS).

4.31.1 Device Name Requirement

V7.3-2

Volume Shadowing for OpenVMS supports device names whose *ddc* portion of the full device name of $*alloclass*$*ddcu*: is three characters.

Prior to this release, it was possible to create device names whose *ddc* portion of the full device name was longer, such as 1DECRAM10:, and these devices mounted successfully. However, mounting such devices as part of a shadow set caused operational problems, such as %MOUNT-F-XSMBRS errors when other disks were added to the shadow set.

Starting with OpenVMS Alpha Version 7.3-2, the Mount utility enforces this three-character requirement for the *ddc* portion of the full device name during the initial attempt to mount the device. If you attempt to mount a device whose name does not conform to this requirement, the following error message is displayed:

```
MOUNT-F-NOTSHDWDEV, not a valid shadow set member
```

4.31.2 Warning About Using SET SHADOW and SHOW SHADOW in DCL Command Procedures

V7.3-2

The new DCL commands SET SHADOW and SHOW SHADOW will continue to evolve. In a future release, the default display and implementation of a SHOW SHADOW/FULL display will change the current formatting. Therefore, HP advises customers not to rely on parsing the current format of output in DCL command procedures to obtain information about the shadow set. Instead, consider using the F$GETDVI lexical function to obtain many of the items displayed by the SHOW SHADOW command.

Furthermore, the behavior of the SET SHADOW command will also change. In addition to other new qualifiers, a new /ALL qualifier will be required if SET SHADOW is used to set characteristics in all shadow sets on a system at the same time.

Please keep these changes in mind if you are writing DCL command procedures that use these new commands.

4.31.3 Write Bitmaps and Dissimilar Device Shadowing (DDS) Caution

V7.3-2

An interaction occurs between write bitmaps and dissimilar device shadowing (DDS) when Volume Shadowing for OpenVMS is used.

DDS, a new feature in OpenVMS Version 7.3-2, allows you to construct shadow sets of disk devices that are of dissimilar sizes. (For details about DDS, refer to the *HP OpenVMS Alpha Version 7.3–2 New Features and Documentation Overview* manual and *HP Volume Shadowing for OpenVMS*.)

Write bitmaps keep track of application writes made to a shadow set virtual unit so that a member can be returned to that virtual unit without the overhead of a full copy. A write bitmap is created when the user issues a DISMOUNT/POLICY=MINICOPY command for a shadow set member or mounts a shadow set using the MOUNT/POLICY=MINICOPY command. When this bitmap is created, its size depends on the current size of the volume.

When a shadow set is mounted, the logical size of the shadow set virtual unit is set to the size of the smallest member unit. When a member of the shadow set is removed, the logical size of the virtual unit is recomputed based on the sizes of the remaining members of the set. Consequently, the logical size of the virtual unit may increase.

When a write bitmap is created for a shadow set, its size is determined by the current size of the shadow set virtual unit. If the virtual unit's size subsequently increases, the bitmap will not cover the entire virtual unit. If the bitmap is then used to bring back a shadow set member with a minicopy operation, the portion of the virtual unit that is not covered by the bitmap will be copied with a full copy operation.

The following example illustrates this problem:

- Shadow set DSA1: consists of these three members:

  1DGA20: (18 GB)
  1DGA21: (36 GB)
  1DGA22: (36 GB)

- 1DGA22: is removed from the shadow set with a minicopy bitmap using the following command:

  ```
  $ DISMOUNT/POLICY=MINICOPY $1$DGA22:
  ```

  The write bitmap is sized for 18 GB, the current size of the shadow set virtual unit.

- 1DGA20: is removed from the shadow set. To allow the file system to utilize the entire 36 GB of the remaining member, use the following command:

  ```
  $ SET VOLUME/SIZE DSA1
  ```

  1DGA20 can no longer be used in this shadow set because it is smaller than the new volume size.

- 1DGA22: is returned to the shadow set using this command

  ```
  $ MOUNT/SYSTEM DSA1:/SHADOW=$1$DGA22: label
  ```

  The logical size of DSA1: remains at 36 GB; however, the bitmap covers only the first 18 GB.

- The first 18 GB of 1DGA22: are copied using the minicopy bitmap; the remaining 18 GB are copied using a full copy operation.

If the removal of a smaller shadow set member is planned, removing it before removing a larger member with a minicopy bitmap will cause a larger bitmap to be created and will avoid the performance impact of a short bitmap. (In the preceding example, you would remove 1DGA20: before removing 1DGA22:.)

4.31.4 KZPDC (Smart Array 5300) Restrictions

V7.3-2

Volume Shadowing for OpenVMS can be used with the KZPDC controller (Smart Array 5300) subject to the restriction that all shadow set members are formed using devices composed of fault-tolerant devices, such as the following:

- RAID 1, also known as controller-based mirroring
- RAID 5, which is striping with parity
- RAID ADG (Advanced Data Guarding), which is striping with multiple parity devices

A fault-tolerant device on the KZPDC (Smart Array 5300) controller is one that can repair data errors when the media yields errors on one of the underlying LUNs.

OpenVMS Alpha Version 7.3-2 and higher supports shadow sets with members whose total block count varies. This new feature is known as dissimilar device shadowing (DDS). DDS allows a KZPDC device to be shadowed with a device from any supported controller.

For all prior OpenVMS versions, all devices must report the same number of total blocks for HBVS to create a multiple-member shadow set. The configuration utility sets the total number of blocks on a KZPDC or MSA1000 device to the closest match that it can make to the requested size. Because the KZPDC and the MSA1000 use the same calculation, a device created on both with the same requested size will be set to the same size. This allows HBVS to create multiple-member shadow sets.

Caution

There are cases where it will not be possible to use HBVS to create a multiple-member shadow set if a fault-tolerant device is not used. For example, a single member shadow set is formed using a device (physical disk or non-fault-tolerant device). If that device subsequently develops nonrecoverable data errors, it will not be possible to use HBVS successfully to add another member to this shadow set. Once the second member is added to the shadow set, HBVS will read the entire source device and copy it to the target device. When the data error is read from the founding or source shadow set member, HBVS will attempt to force all of the current shadow set members (the source member and the copy target) to create a ''bad spot''. If this request to create a bad spot fails on either shadow set member, the shadow set will be reduced to one member.

4.31.5 Changes in Shadow Set Merge Delay Computation

V7.3-2

During an unassisted shadow set merge operation, read I/O performance available to applications is reduced by two factors:

- The need to perform data consistency checks on all read I/Os
- Contention for I/O bandwidth by the shadow set merge operation

The shadow set merge operation employs a throttling mechanism to limit the impact of merge I/O on applications. The merge process is throttled by introducing a delay between merge I/Os when system load is detected. The logic for computing this delay has been redesigned for OpenVMS Alpha Version 7.3-2. With the new merge delay computation, the default parameter settings will result in faster merge rates for some I/O controllers, such as the HSG-80. For more information, refer to the *HP Volume Shadowing for OpenVMS* manual.

4.31.6 ANALYZE/DISK/SHADOW Command Behavior

V7.3-2

When you specify the /SHADOW qualifier (new in OpenVMS Version 7.3-2) with the ANALYZE/DISK_STRUCTURE command, the entire contents of a shadow set or a specified range of blocks in a shadow set are examined for discrepancies.

If a member of the shadow set experiences connectivity problems for any reason, the ANALYZE/DISK_STRUCTURE command displays the error that it received and then returns the user to the DCL prompt.

To correct the connectivity problem and run the utility again on the same shadow set, you might need to create a temporary file on the virtual unit before reissuing the ANALYZE/DISK/SHADOW command.

Additionally, this utility may report explainable discrepancies between the shadow set members if the shadow set has not undergone a full merge since the shadow set was created. This happens if the shadow set was created using the DCL command INITIALIZE/SHADOW without the /ERASE qualifier and if the disk devices had different contents. It is important to realize that this is *not* disk corruption. The blocks that are reported as different have not been written to, but they may contain stale data; the blocks reported as inconsistent may even be allocated to a file because there may be unwritten space between the file's end-of-data location and the end of the allocated space.

You can eliminate such inconsistencies by performing a full merge. To initiate a full merge, execute the DCL command SET SHADOW/DEMAND_MERGE DSA*xxx*. If the devices are served by controllers that support controller-based minimerge (for example, HSJ50s), this command should be issued while the shadow set is mounted on only one node within the cluster. Otherwise, a minimerge will occur, and the discrepancy may not be resolved. When you are adding members to a single member shadow set, a full copy operation will also ensure that the disk is consistent both within and outside of the file system.

If errors are reported on an ANALYZE/DISK/SHADOW command after a full merge has been executed, they should be investigated.

Also see Section 4.31.7 for another note about ANALYZE/DISK/SHADOW command behavior.

4.31.7 ANALYZE/DISK/SHADOW Command Behavior with Dissimilar Device Shadow Sets

V7.3-2

An ANALYZE/DISK/SHADOW command may also report explainable discrepancies if a full merge has not occurred since the shadow set was logically expanded after a new member was added. The following example illustrates this problem:

- Shadow set DSA1: consists of two members:

  1DGA20: (18 GB)
  1DGA21: (36 GB)

- A second 36-GB member, 1DGA22:, is added to the shadow set with a full copy operation.
- After the copy completes, 1DGA20: is removed from the shadow set.
- At this point, if the SET VOLUME/SIZE DSA1: command is executed, the shadow set virtual unit DSA1: will increase to 36 GB. Then, ANALYZE/DISK/SHADOW will report discrepancies because only the first 18 GB of the shadow set contents were copied to 1DGA22:.

The discrepancies reported by ANALYZE/DISK/SHADOW are harmless because the space in question has not yet been written by applications.

Also see Section 4.31.6 for another note about ANALYZE/DISK/SHADOW command behavior.

4.31.8 Dismount of Shadow Set Member Using /MINICOPY

V7.3

In a single site or in a multiple-site OpenVMS Cluster configuration, if you issue a DISMOUNT command with the /MINICOPY qualifier from a client system to dismount a shadow set member, the command might fail.

Workaround

If the first DISMOUNT command fails, repeat the command, as shown in the following example:

```
$! The following commands are NOT executed on the WILD3 system.
$
$ SHOW DEVICE DSA5555
Device                  Device           Error    Volume         Free  Trans Mnt
 Name                   Status           Count     Label        Blocks Count Cnt
DSA5555:                Mounted              0  $80$DKA107:    7994646     1  18
$80$DKA107:    (WILD3)  ShadowSetMember      0  (member of DSA5555:)
$80$DKA302:    (WILD3)  ShadowSetMember      0  (member of DSA5555:)
$80$DKA303:    (WILD3)  ShadowSetMember      0  (member of DSA5555:)
$
$
$ DISMOUNT/POLICY=MINICOPY $80$DKA302:
%DISM-W-CANNOTDMT, $80$DKA302: cannot be dismounted
%DISM-F-SRCMEM, only source member of shadow set cannot be dismounted
$
$
$ DISMOUNT/POLICY=MINICOPY $80$DKA302:
$
```

This problem will be corrected in a future release.

5
Programming Release Notes

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

5.1 Programs Must Be Recompiled and Linked (I64 Only)

V8.2

All programs must be recompiled for OpenVMS I64 Version 8.2. This is the case even if you have compiled and linked for an earlier evaluation or field test release of OpenVMS I64. (HP reserves the right to make incompatible changes to evaluation releases.) With this production release of OpenVMS I64 Version 8.2, HP's normal upward-compatibility policy goes into effect.

For information about recompiling Alpha programs, refer to Section 5.2.

5.2 Privileged Programs May Need to Be Recompiled (Alpha Only)

V8.2

OpenVMS Alpha Version 8.2 is a major version release in which a number of privileged data structures have changed. It might be necessary to recompile and relink privileged applications linked with /SYSEXE that refer to internal OpenVMS data structures or routines.

If you get a SYSVERDIF error message when you invoke an image or load a device driver, this indicates that the privileged image or driver was compiled and linked under a prior version of the operating system. You must then recompile and relink the image or driver to run on OpenVMS Alpha Version 8.2.

For information about recompiling I64 programs, refer to Section 5.1.

5.3 Privileged Data Structures Updates

V8.2

OpenVMS Version 8.2 contains updates for a number of privileged data structures. These changes apply to both Alpha and I64 systems. The majority of these data structure updates are to support future scaling and performance projects in the operating system. As a result of these changes, any images or drivers that link against the base operating system (that is, those that use /SYSEXE on the LINK command) might need to be recompiled and relinked in order to run on OpenVMS Version 8.2.

The privileged data structure changes do not necessarily affect all privileged images and drivers — only those linked with one of the specific subsystems that has changed. For these subsystems, the major version identification number associated with the subsystem has been increased. The subsystems that have changed are the following:

SYS$K_IO

SYS$K_MEMORY_MANAGEMENT
SYS$K_CLUSTERS_LOCKMGR
SYS$K_FILES_VOLUMES
SYS$K_CPU
SYS$K_MULTI_PROCESSING
SYS$K_PROCESS_SCHED

Note: On I64 systems, these versions are reported as SYS$K_VERSION_*xxxx*.

The versions of these subsystems are linked into images based on the usage of various privileged system routines and data cells. You can use the ANALYZE/IMAGE utility to determine with what specific subsystem a privileged image is linked. For example:

```
$ ANALYZE/IMAGE IMAGE.EXE /OUTPUT=IMAGE.TXT
$ SEARCH IMAGE.TXT "SYS$K_"
```

If any of the versions reported match this list, OpenVMS Version 8.2 will fail to activate the image and issue a SS$_SYSVERDIF (system version mismatch error) for images linked on prior versions of the operating system.

5.3.1 KPB Extensions

V8.2

Prior versions of OpenVMS supported the use of KPBs for kernel mode above IPL 2. To make the transition to I64 easier, usage of KPBs has been expanded for use in outer modes and all IPLs. This Alpha and I64 change allows certain code that previously had private threading packages to make use of KPBs on both Alpha and I64. In order to support these changes to kernel processes, some changes to the KPB structure were required. No source changes should be necessary for existing Alpha code.

5.3.2 CPU Name Space

V8.2

OpenVMS currently has an architectural limit of a maximum CPU ID of 31. Various internal data structures and data cells have allocated 32 bits for CPU masks. We are increasing the amount of space allocated for these masks to 64 bits for Alpha and 1024 bits on I64 to allow supporting larger CPU IDs in a future release. The existing longword CPU mask symbols and data cells will still be maintained.

There should be no initial impact to privileged images and drivers. However, at some time in the future, HP will document how privileged products that refer to CPU masks must change their code to support systems with CPU IDs greater than 31.

5.3.3 64-Bit Logical Block Number (LBN)

V8.2

OpenVMS today supports LBNs of only 31 bits. This limits our support of a disk volume to 1 terabyte. The amount of space allocated for internal LBN fields is being increased to 64 bits to allow support for larger disk volumes in the future. The existing longword LBN symbols will still be maintained and will be overlaid with a quadwords symbol.

5.3.4 Forking to a Dynamic Spinlock

V8.2

In order to scale the OpenVMS operating system on large SMP systems, a number of areas in the operating system have been using dynamic spinlocks as opposed to the very limited number of static spinlocks. The ability to FORK and have the fork dispatcher obtain synchronization with a dynamic spinlock is desirable. We are adding this capability to OpenVMS Version 8.2 by extending the size of the FKB structure and by adding a FKB$L_SPINLOCK field to the end of this structure. This spinlock field is referenced only if FKB$B_FLCK contains the value SPL$C_DYNAMIC. The FKB structure is embedded in many other system data structures, and this change impacts the size and layout of a large number of privileged data structures

Applications that copy the FKB$B_FLCK field from an OpenVMS created structure to another FKB should also consider copying the data in the FKB$L_ SPINLOCK field.

HP recommends that privileged code check for cases of allocating FKB structures and using a hard-coded value for the size of 32. Code should use the symbol FKB$C_LENGTH for the size of a FKB.

5.3.5 UCB/DDB Updates

V8.2

A number of updates have been made to the UCB and DDB structures.

The list of UCBs associated with a DDB is currently a singularly linked list. When creating and deleting a UCB, this list must be walked until the appropriate location is found. For OpenVMS Version 8.2, UCBs are now linked to the DDB with a double linked list. In addition, the DDB maintains a seed pointer to where the search should start when creating a new unit to allow for faster device creation. Drivers that manipulate their unit seed pointer in a template UCB will not be able to take advantage of the faster device creation.

Any code that manipulates the DDB list of UCBs will no longer work correctly. HP highly recommends that you use the provided internal routines for linking and unlinking UCBs. Code-walking the list of UCB forward continues to work correctly.

The UCB$W_UNIT field is currently a 16-bit word field. There are now 32 bits allocated for this field. The UCB$W_UNIT field will still be maintained, so no source code changes are necessary. In a future release, OpenVMS might support larger unit numbers. This would be done only for drivers that indicate they can support this feature.

Byte and Word fields in the terminal driver's UCB extension are now aligned on longword boundaries.

5.3.6 PCB$T_TERMINAL Size Increase

V8.2

The Process Control Block (PCB) structure contains a field PCB$T_TERMINAL, which is 8 bytes to hold the device name for an interactive process (such as LTA123:, RTA7:, NVA456: and so forth). This field is a counted ASCII string, with the first byte being the length of the string and the remaining 7 bytes holding the device name. With a 3-letter device name, only four digits can be used to hold the unit number, and the colon would be stripped off for unit

numbers greater than 999. For OpenVMS Version 8.2, this field has been increased to 16 bytes to hold device names with larger unit numbers.

If you fetch this field using a call to $GETJPI with the JPI$_TERMINAL item code, you are not impacted, but you might want to increase the buffer passed to the system service to hold up to 16 bytes.

5.3.7 Per-Thread Security Impacts Privileged Code and Device Drivers

V7.3-1

The method used for attaching a security profile to an I/O Request Packet (IRP) changed with Version 7.2.

In versions of OpenVMS prior to Version 7.2, the IRP structure contained the address of the processwide Access Rights Block (ARB) security structure of the requestor. Beginning with OpenVMS Alpha Version 7.2, the address of the new security profile structure (Persona Security Block, or PSB) was added to the IRP as a functional replacement of the ARB address.

The I/O subsystem maintains its access to the PSB through a reference counter within the PSB. The I/O subsystem increments this reference counter at the time of IRP creation and decrements the counter at I/O postprocessing of that IRP. When this counter reaches zero, the PSB structure is deallocated.

Device drivers that create or clone copies of IRPs to facilitate multiple I/O operations per request, and subsequently pass the copies to the I/O subsystem for postprocessing, must make code changes to account for the extra references to the PSB in these additional IRPs. This is done by passing the PSB address located in the copied IRP to the NSA_STD$REFERENCE_PSB routine. The include file and routine call for NSA_STD$REFERENCE_PSB is as follows:

```
#include <security-macros.h>

/* Increment REFCNT of PSB that is now shared by both IRPs  */

nsa_std$reference_psb( irp->irp$ar_psb );
```

Device drivers need to make this change under the following conditions:

- If a device driver creates a new IRP by duplicating an existing IRP and submits both the original and the duplicate IRPs for I/O postprocessing by calling IOC_STD$SIMREQCOM or IOC_STD$DIRPOST1, the device driver must call NSA_STD$REFERENCE_PSB sometime after duplicating the IRP, but before submitting it for I/O postprocessing.
- If a device driver creates a new IRP by duplicating an existing IRP and does not put the address of some procedure descriptor into the IRP$L_PID cell in either the copy or the original IRP, and the device driver submits both the original and the duplicate IRPs for I/O postprocessing by calling IOC_STD$REQCOM, COM_STD$POST, COM_STD$POST_NOCNT, or IOC_STD$POST_IRP, the device driver must call NSA_STD$REFERENCE_PSB sometime after duplicating the IRP but before submitting it for I/O postprocessing.

  Device drivers that perform these steps are also likely to put the address of some procedure descriptor into IRP$L_PID. Therefore, most device drivers that duplicate IRPs should be able to function correctly on OpenVMS Version 7.2 or higher without making source changes, relinking, or recompiling.

Failure to call NSA_STD$REFERENCE_PSB in these circumstances will result in corrupt tracking information within the PSB, which can result in system failures.

If you make code changes in a device driver to call NSA_STD$REFERENCE_ PSB, you must recompile and relink the driver to run on OpenVMS Version 7.2 or higher.

5.3.8 IPL Requirement For OpenVMS Fork Thread Creation Now Enforced

V7.3-1

Several routines are used by privileged code to create OpenVMS fork execution threads. These routines run in system context independent of any process. There are four variations of these routines, depending on whether an immediate or queued fork is required and on which language interface is being used:

- EXE$QUEUE_FORK
- EXE_STD$QUEUE_FORK
- EXE$PRIMITIVE_FORK
- EXE_STD$PRIMITIVE_FORK

These routines must be called at or above IPL$_RESCHED, to prevent accidental rescheduling to a different CPU during their execution. Such a reschedule could cause the system to hang.

In OpenVMS V7.3-1, if SYSTEM_CHECK is set to 1, these routines check the system IPL at entry. If the IPL is below IPL$_RESCHED, the system will fail with an SPLINVIPL bugcheck.

For performance reasons, the IPL is not verified if SYSTEM_CHECK is set to zero (the default). Incorrect code may cause the system to hang if a reschedule to another CPU occurs during execution of these routines from process context (for example, below IPL$_RESCHED).

5.4 Applications Using Floating-Point Data

V8.2

The Itanium® architecture implements floating-point arithmetic in hardware using the IEEE floating-point formats, including IEEE single and IEEE double.

The Alpha hardware supports both IEEE and VAX floating-point formats. On OpenVMS Alpha, the compilers generate code to use the VAX formats by default with options to use the IEEE formats.

On OpenVMS I64, the compilers generate code to use the IEEE formats by default with options to use the VAX formats. HP recommends the use of IEEE formats on Integrity servers unless applications need to process VAX floating-point binary data that has been generated on VAX or Alpha systems. For details about using VAX formats on OpenVMS I64, refer to the OpenVMS Floating-Point White Paper on the following website:

`http://www.hp.com/products1/evolution/alpha_retaintrust/openvms/resources.html`

OpenVMS Version 8.2 contains new IEEE floating-point versions of LIB$ and OTS$ run-time library routines for use by applications using IEEE floating-point data. The function prototype definitions of these new routines are not included in the system-supplied header files. To use these functions you must define the function prototypes in your applications. Refer to the LIB$ and OTS$ run-time library reference manuals for descriptions of these functions. These function prototype definitions will be included in the system-supplied header files in a future release.

5.5 Ada Compiler Not Yet Available (I64 Only)

V8.2

The Ada compiler is supported on OpenVMS Alpha Version 8.2. HP is not porting the HP Ada 83 compiler from Alpha to I64; AdaCore is porting the Ada 95 compiler to OpenVMS I64. Customers can contact AdaCore directly when this product becomes available.

5.6 Backup API: Journaling Callback Events Restriction

Permanent Restriction

If an application registers a callback routine for any of the journaling events, it must register a callback routine for all the journaling callback events. The following is a list of the journaling callback events:

BCK_EVENT_K_JOURNAL_OPEN
BCK_EVENT_K_JOURNAL_WRITE
BCK_EVENT_K_JOURNAL_CLOSE

Refer to the Backup API chapter in the *OpenVMS Utility Routines Manual* for more information about registering callback routines.

5.7 C Programs: Compiling with CASE_LOOKUP=SENSITIVE Settings

Permanent Restriction

If you are compiling C programs in a process where the characteristics were set to CASE_LOOKUP=CASE=SENSITIVE, any #include files in your C program specified with the .h file type (lowercase h) will not be seen and executed. In addition, if a system #include file specifies another #include file with a .h file type, the second #include file will not be seen and an error will be generated.

To avoid this behavior, compile with case set to blind. If it is necessary to use `case=sensitive`, specify any #include files in your C programs either with no file type (for example, `#include <stdio>`) or with an uppercase H file type (for example, `#include <stdio.H>`).

Note that this does not correct the scenario where system #include files, such as stdlib.h, in turn specify #include files with a .h file type and cause an error to be generated.

5.8 C Run-Time Library

The following sections describe changes and corrections to the C Run-Time Library (RTL).

5.8.1 Memory Leak in Programs Using socket_fd Fixed

V8.2

Previously, there was a memory leak in programs using `socket_fd`, which in certain circumstances consumed the page-file quota. This problem has been fixed.

5.8.2 vsnprintf and snprintf User Buffer Overwrite Fixed

V8.2

Previously, under certain conditions, the `vsnprintf` and `snprintf` functions could overwrite memory beyond the maximum count allowed in the user's buffer.

For example, the following call to `snprintf` would overwrite user buffer `t` beyond the allowed `t[2]`, using the format string provided:

```
snprintf(t, 3, "%2sxxxx", "a");
```

This problem has been fixed.

5.8.3 mmap and mprotect Changes

V8.2

Previously, a change in the C RTL for OpenVMS Versions 7.3-1 and 7.3-2 would return an error unless memory for the `mprotect` function had already been mapped by `mmap`.

This problem has been fixed by restoring the legacy behavior: you can once again set protection using `mprotect` for memory that was not mapped by `mmap`.

5.8.4 getpwnam_r and getpwuid_r Pointer Problem Fixed

V8.2

Previously, programs calling the short-pointer version of `getpwuid_r` (`_getpwuid_r32`) could incorrectly pass a long-pointer value for the third argument (*buffer*).

This user-specified buffer was allocated to members of the resulting `passwd` structure, which are 32-bit pointers. This caused an incorrect result for those members (`pw_name`, `pw_dir`, and `pw_shell`) if the buffer was in high memory.

The same problem existed for the short-pointer version of `getpwnam_r` (`_getpwnam_r32`).

This problem has been fixed. The prototypes for `_getpwnam_r32` and `_getpwuid_r32` have been changed so that the functions now accept only a 32-bit pointer for the *buffer* argument instead of allowing a 64-bit pointer.

5.8.5 _strtok_r32 and _strtok_r64 Now in Scope

V8.2

Previously, programs that included `<string.h>` and called `_strtok_r32` or `_strtok_r64` would not find a prototype in scope. This problem has been fixed.

5.8.6 const Type Qualifier Added to iconv Prototype (Alpha Only)

V8.2

To conform to the X/Open standard, the `const` qualifier will be added to the second argument of the `iconv` function prototype in `<iconv.h>` when the XOPEN_SOURCE feature test macro is defined to be 500 (#DEFINE XOPEN_SOURCE 500):

```
size_t iconv (iconv_t cd, const char **inbuf, size_t *inbytesleft,
              char **outbuf, size_t *outbytesleft);
```

5.8.7 <stddef.h> Header Modified (Alpha Only)

V8.2

Previously, the C++ compiler did not recognize the result of the `offsetof` macro as an integral constant expression, which led to C++ compiler errors when used in that manner.

This problem has been fixed. The `<stddef.h>` header file has been modified to provide an alternative definiton of the `offsetof` macro for use by C++ Version 6.5 and later.

This alternative definition uses an EDG extension (__INTADDR__) to perform the otherwise nonstandard conversion from pointer to integer. This solution was provided by and recommended by EDG, and is the same as their solution (except for the use of nonpolluting names).

5.8.8 getc Macro Argument Now Protected by Parentheses (Alpha Only)

V8.2

To comply with the ISO C Standard (ISO/IEC 9899), the argument to the `getc` macro is now protected by parentheses.

5.8.9 CXXL Prefix Problems with Inlined Functions getc and getchar Fixed (Alpha Only)

V8.2

Previously, C++ compilations of `getc` and `getchar` resulted in undefined symbol warnings at link time. Under certain circumstances, the `getc` macro (in C) or inline function (in C++) called the actual `decc$getc` function. The problem was that the inline `getc` function declared the actual `decc$getc` function under extern_prefix "CXXL$". A similar problem occurs with `getchar`.

This problem has been fixed. The `<stdio.h>` header file has been modified to provide a prototype for `getc` and `getchar` within the scope of the `#pragma__extern_prefix "CXXL$"` directive, while leaving the inlined implementation definition outside the scope of CXXL$ prefixing.

5.8.10 Non std Functions Declared in std Namespace (Alpha Only)

V8.2

Previously, C++ compilations of source files that reference `v*scanf` or `*snprintf` functions resulted in %CXX-E-UNDECLARED errors because the `<stdio.h>` header declared these functions in the `std` namespace, but did not inject them into the global namespace.

This problem has been fixed. These function declarations have been moved to a location outside the `std` namespace.

5.8.11 lseek on Large File Offset Problem Fixed (Alpha Only)

V8.2

Previously, the `lseek` function failed to correctly position files on offsets larger than ULONG_MAX (4294967295) bytes (under the _LARGEFILE feature control macro). For example, calling `lseek` on a 6-gigabyte (16 megablock) file and specifying an offset of 0x100000000 left the file position at 0. This problem has been fixed.

5.8.12 New EABANDONED Code in <errno.h>

V8.2

A new `errno` code EABANDONED has been added to the `<errno.h>` header file.

Pthreads functions can now return an EABANDONED ("Owner cannot release resource") code if the system has determined that the target process-shared mutex is locked by a process that has terminated (that is, the mutex is considered "abandoned" because the rightful owner cannot release it).

5.8.13 mktime Problem Fixed

V8.2

Previously, the UTC-time-based function `mktime` lost a day when the structure member `tm_mday` was supplied with 0 or negative values. `mktime` also generated inconsistent days. This problem has been fixed.

5.8.14 POLLWRNORM Now the Same as POLLOUT in <poll.h>

V8.2

According to the X/Open documentation for the `<poll.h>` header file, POLLWRNORM should equal POLLOUT. Previously it did not; now it does.

5.8.15 IPV6 Structures in <in6.h> Now Packed

V8.2

The IPV6 structure `sockaddr_in6` in the `<in6.h>` header file was not packed, causing problems when applications expected and checked the size for packed structures. This structure should have been packed because the member alignment is on natural boundaries. This problem has been fixed.

5.8.16 __PAL_BUGCHK Fixed in <builtins.h>

V8.2

Previously, when using the C and C++ compilers, calling __PAL_BUGCHK with a parameter resulted in a fatal error. Also, many of the builtins had misleading comments about their implementation. This has been fixed.

5.8.17 C++ Compiler Error with statvfs Fixed

V8.2

The `restrict` type qualifier is added to the `<decc$types.h>` and `<statvfs.h>` header files when the C or C++ compiler or the X/Open standard (XPG6) supports it.

This fixes the following problem:

```
int statvfs(const char * __restrict path, struct statvfs * __restrictbuf);
..................................^
%CXX-E-EXPRPAREN, expected a ")"
```

5.8.18 glob and globfree Issues Fixed

V8.2

The following `glob` and `globfree` issues have been fixed:

- Previously, the `glob` and `globfree` functions did not support 64-bit pointers. Passing 64-bit pointers to the C RTL could create unpredictable behavior (such as accvios) because the C RTL did not expect 64-bit pointers and the data structure for `glob` supported only 32-bit pointers.
- The `<glob.h>` header file did not specify an alignment pragma. Customers could set `member_alignment` around their include of `<glob.h>` and cause type `glob_t` to be incorrectly aligned, possibly causing the C RTL to behave unpredictably.

5.8.19 DECC$SHR_EV56 Now Linked Correctly

V8.2

Previously, OpenVMS Version 7.3-2 did not correctly optimize the DECC$SHR_EV56.EXE image.

The C RTL build is done with two sets of objects: one compiled normally, and one optimized for Alpha EV56 processors. The DECC$SHR_EV56.EXE image was not linked with the optimized objects. This problem has been fixed.

5.8.20 Zone Information Compiler (zic) Updates

V8.2

New time indicators have been added for the AT field in the Rule line.

The letter "u" (or "g" or "z") indicates that the time in the AT field is UTC.

For details about `zic`, refer to the *HP C Run-Time Library Reference Manual for OpenVMS Systems*.

5.9 Calling Standard and Rotating Registers (I64 Only)

V8.2

This note supplements information in the *HP OpenVMS Calling Standard*.

The Calling Standard invocation context block (ICB) (see Table 4-16 in the *HP OpenVMS Calling Standard*) and mechanism vector (see Figure 8-7 and Table 8-6 in the *HP OpenVMS Calling Standard*) always record general, floating-point, and predicate registers as if the register rename base (CFM.rrb) and rotating size (CFM.sor) were both zero. In other words, when rotating registers are in use, the effects of the rotation are ignored. This is also true of the register masks used by the LIB$I64_PUT_INVO_REGISTERS routine (see Section 4.8.3.13 in the *HP OpenVMS Calling Standard*), because these masks are defined in terms of fields in the ICB structure.

Currently, the supplemental access routines LIB$I64_GET_FR, LIB$I64_SET_FR, LIB$I64_GET_GR and LIB$I64_SET_GR (see Section 4.8.4 in the *HP OpenVMS Calling Standard*) improperly interpret their register number parameters without applying an adjustment for the effects of the register rename base and rotating size registers. This is an error and will be corrected in a future release.

In the meantime, any code that examines the general, floating-point, and predicate registers in an ICB or mechanism vector and that seeks to interpret the register contents as it would be seen by the code during its execution must examine the saved CFM register and apply the appropriate adjustments itself.

5.10 Common Data Security Architecture (CDSA) Considerations

The following notes pertain to CDSA.

5.10.1 Secure Delivery Advanced Developer's Kit

V8.2

Downloading of files over the Internet is becoming a requirement of OpenVMS customers who want to use this easy and convenient method to acquire software updates, but are wary of the security vulnerabilities. Secure Delivery makes use of public key and digital signature technology to implement a system that provides OpenVMS users with the ability to authenticate and validate the files they download from OpenVMS and third party OpenVMS vendors.

The preliminary documentation for Secure Delivery can be found on the following website:

`http://www.hp.com/go/openvms/security/`

This documentation provides an overview of Secure Delivery on OpenVMS, and how to invoke its components. Secure Delivery is provided as an Advanced Developer's Kit (ADK) with OpenVMS Version 8.2, and is planned to be fully supported in a subsequent release of the operating system. With the exception of a future tie-in to PCSI for automatic verification of software kits, all of the Secure Delivery functionality is present in OpenVMS Version 8.2.

5.10.2 Installation and Initialization Considerations

V7.3-2

Installation of CDSA is done automatically when you install the operating system. However, you must be aware of the following considerations:

- Although the CDSA installation is part of the OpenVMS Version 7.3-2 (and later) installation, the setup and initialization of CDSA must be performed separately. Before you can use CDSA, you must perform the following manual procedure. You need the SYSPRV privilege to run this procedure. Enter the following command:

  ```
  $ @SYS$STARTUP:CDSA$INITIALIZE
  ```

  When a new version of CDSA is installed (for example, in an upgrade from field test to a production version, or to a new version of OpenVMS), you must run the CDSA upgrade procedure (@SYS$STARTUP:CDSA$UPGRADE). You should shut down any CDSA application before you run the upgrade procedure.

  It is not necessary to rerun the initialization or upgrade procedures when the system is rebooted. You also do not need to add the initialization or upgrade procedures to the OpenVMS startup procedures.

- Do not attempt to remove CDSA from your system. Use of the PCSI command PRODUCT REMOVE is not supported for CDSA, even though there is an apparent option to remove CDSA. (This option is due to the use of PCSI in the installation.) CDSA is installed together with the operating system and is tightly bound with it. An attempt to remove it from OpenVMS will not work cleanly and could create other undesirable effects. An attempt to remove CDSA will result in a message similar to the following:

```
%PCSI-E-HRDREF, product CPQ AXPVMS CDSA Vn.n is referenced
   by DEC AXPVMS OPENVMS V8.2
-PCSI-E-HRDRF1, the two products are tightly bound by this
   software dependency
```

5.11 CONVERT-I-SEQ Error on CONVERT/NOSORT with Collated Key

V7.3

This potential change in behavior is restricted to a CONVERT command that specifies both the /NOSORT qualifier and a collated key type on one of the keys in the output file.

The /NOSORT qualifier on a CONVERT command indicates that the primary key is already in sorted order in the input file and directs the Convert utility not to sort it. Prior to OpenVMS Version 7.3, the Convert utility had a defect that caused it to always sort the input file if some key specified for the output file had a collated key type, regardless of whether /NOSORT was specified. As of OpenVMS Version 7.3, the Convert utility has been fixed to appropriately obey the /NOSORT qualifier on the command line, even if one of the keys in the output file is a collated key.

This means that a convert operation that previously succeeded as a side-effect of the collated key defect may now produce %CONVERT-I-SEQ messages if the input file is not already in sorted order by the primary key and /NOSORT is specified on the command line. The /NOSORT qualifier should not be used if the input file is not already in sorted order by the primary key.

5.12 Debugging Modes: Avoiding CPUSPINWAIT Bugchecks

V7.3-1

The OpenVMS operating system has a number of special modes of operation designed to help you debug complex hardware and software problems. In general terms, these special modes enable an extra level of tracing, data recording, and consistency checking that is useful in identifying a failing hardware or software component. These modes of operation are controlled by several system parameters: MULTIPROCESSING, POOLCHECK, BUGCHECKFATAL, and SYSTEM_CHECK.

If you are using one of these special modes (for example, to debug a device driver or other complex application), under certain conditions, generally related to high I/O loads, it is possible to incur a CPUSPINWAIT bugcheck. Specifically, any privileged code that runs for extended periods of time while holding a spinlock can cause a CPU spinwait bugcheck. Spinlocks are used to delineate the entry and exit points for critical sections, and should not be held continuously, as can occur in this situation.

To prevent a CPUSPINWAIT bugcheck, either use the system default settings for these system parameters, or reduce the loading of the system.

If you have reason to change the default settings, you can reduce the likelihood of encountering a problem by setting the SMP_LNGSPINWAIT system parameter to a value of 9000000.

5.13 Delta/XDelta Debuggers

The following release notes pertain to the OpenVMS Delta and XDelta debuggers running on OpenVMS Alpha and I64 systems.

OpenVMS Debugger release notes are in Section 5.28.

5.13.1 Delta Debugger Not Available on OpenVMS I64

V8.2

The Delta debugger has not yet been ported to the OpenVMS I64 operating system.

5.13.2 Restrictions for XDELTA on I64 Systems

V8.2

The following Intel® Itanium® hardware registers are not supported by XDELTA:

- CPUID
- Debug Data Break Registers
- Debug Instruction Break Registers
- Region Registers
- Protection Key Registers
- Instruction Translation Registers
- Data Translation Registers
- Device Interrupt Control Register

5.13.3 Differences Between XDELTA on I64 Systems and Alpha Systems

V8.2

To interrupt a running system on OpenVMS I64, press Ctrl/P at the system console. Note that XDELTA must have been loaded previously. When you press Ctrl/P, the system is halted at the current PC and at the current IPL. There is no delay in waiting for the IPL to drop below 14, as there is on OpenVMS Alpha systems.

5.13.4 XDELTA Register Display Consideration (I64 only)

V8.2

XDELTA on OpenVMS I64 displays general, floating-point, and predicate registers as if the register rename base (CFM.rrb) and the rotating size (CFM.sor) are both zero. In other words, when rotating registers are in use, the effects of the rotation are ignored. This condition will be corrected in a future release. See Section 5.9 for additional details.

5.14 File Applications: Corrections to *Guide to OpenVMS File Applications*

V8.2

The following corrections will be made to the *Guide to OpenVMS File Applications* if this manual is revised in the future.

- Table 1-4 is potentially confusing. In the comparison of file formats, directory limitations are listed as 256 for ODS-2 and ODS-5 file formats. This statement should be clarified to say **256 directory levels** so that users do not think directories are limited to 256 files.

  A similar table in the *HP OpenVMS System Manager's Manual* has been corrected for Version 8.2.

- In Section 6.6.3, replace the fourth paragraph (not counting the note) with the following text:

  "When you define the rooted-device logical name for use in a program in a SET DEFAULT command, you specify that the logical name is a concealed-device logical name by using the /TRANSLATION_ ATTRIBUTES=CONCEALED qualifier with the DCL command DEFINE or ASSIGN. To define the concealed-device logical name as a rooted-device logical name, the root directory must contain a trailing period (.), such as DUA22:[ROOT.] and the device specification must be a physical device name. The equivalence name for a rooted-device logical name must not contain other logical names. When specifying a directory, you can use only a trailing period for the root directory."

5.15 HP BLISS Compiler Warnings with RMS Structures (I64 Only)

V8.2

Quadword alignment has been added to the BLISS macros ($*xxx*_DECL) that can be used to allocate RMS user structures (for example, FAB, RAB). Alignment faults are costly to performance—especially as processors get faster. By implementing the alignment directly in the macros, a number of OpenVMS utilities and user applications written in BLISS that use these macros show improved performance.

The specific names of the macros are: $FAB_DECL, $NAM_DECL, $NAML_ DECL, $RAB_DECL, $RAB64_DECL, $XABALL_DECL, $XABDAT_DECL, $XABFHC_DECL, $XABITM_DECL, $XABJNL_DECL, $XABKEY_DECL, $XABPRO_DECL, $XABRDT_DECL, $XABRU_DECL, $XABTRM_DECL, and $XABSUM_DECL.

The alignment added in the RMS macros might result in compiler warnings of conflicting alignment. Programs with compiler warnings should link and execute correctly. However, the minor source changes to eliminate the warnings are recommended.

If you use any of these macros in a BLISS application and the declaration includes the ALIGN attribute, the BLISS compiler issues a ''conflicting or multiply specified attribute'' warning. For example, the warning is issued for the following declaration: FAB: $FAB_DECL ALIGN(2). The compiler also issues this warning even if quadword alignment (ALIGN(3)) is specified. You should remove any explicit ALIGN attributes associated with these macros.

In addition, if any of these allocations is included in a PSECT that contains an explicit alignment that is in conflict with ALIGN(3) (that is, is lower than ALIGN(3)), the BLISS compiler issues an ''align request negative or exceeds that of psect'' warning. For example, the warning would be issued for the following declaration:

```
PSECT OWN = $OWN$ (..., ALIGN(2), ...)

OWN

    FAB = $FAB_DECL, ...
```

If warnings on PSECT alignment are seen when recompiling a BLISS application, the correction is to increase the alignment of the PSECT to ALIGN(3) (or higher). In rare cases, applications may have assumptions on adjacency of data between PSECTs. Those assumptions could be broken in making this change. Therefore, check the code for such assumptions and make any necessary corrections.

While a number of OpenVMS utilities are written in BLISS, only a few warnings were generated in a full OpenVMS build. Changes in OpenVMS to eliminate warnings did not require other changes to correct assumptions. Based on this, few user applications likely will require modification.

5.16 HP COBOL Run-Time Library (RTL)

V8.2

For OpenVMS Alpha and OpenVMS I64 Version 8.2, the HP COBOL RTL (DEC$COBRTL) has been updated to V2.8-775.

5.16.1 SET and COB$SWITCHES

V8.2

The implementation of the COBOL SET verb (implemented using the logical COB$SWITCHES) has been changed to use LNM$PROCESS if the attempt to use the first entry in LNM$FILE_DEV fails. Previously, the COBOL SET verb failed if the first entry in LNM$FILE_DEV was a nonwritable logical name table.

5.16.2 Record Locking Problem Corrected

V8.2

Under certain circumstances with the START or WRITE statement, COBOL used to acquire multiple record locks with automatic record locking. This problem has been fixed.

5.17 HP Decimal Support Run-Time Library (RTL)

V8.2

For OpenVMS Alpha and OpenVMS I64 Version 8.2, the HP Decimal Support RTL (LIBOTS2) has been updated to V2.8-67.

5.18 HP Fortran for I64

V8.2

The OpenVMS I64 Fortran compiler is a port of HP Fortran 90 for OpenVMS Alpha. It runs on OpenVMS I64 systems and produces objects for OpenVMS I64 systems. The objects are linked using the standard linker on OpenVMS I64. This compiler requires OpenVMS I64 Version 8.2.

HP Fortran for OpenVMS I64 systems features the same command-line options and language features as HP Fortran 90 for OpenVMS Alpha systems with the following exceptions:

- Floating-point arithmetic

  Refer to the white paper *OpenVMS Floating-Point Arithmetic on the Intel® Itanium® Architecture* for essential reading. You can link to this document from the following web site:

  `http://h71000.www7.hp.com/openvms/integrity/resources.html`

- IEEE is the default floating-point datatype (that is, the default is /FLOAT=IEEE_FLOAT).
- The /IEEE_MODE qualifier defaults to /IEEE_MODE=DENORM_RESULTS.
- Users should pick one /FLOAT value and one /IEEE_MODE value and keep it for the whole application.
- Only the F90 compiler is supported. The F77 compiler, previously invoked with the /OLD_F77 qualifier, is not available. Some of the functionality contained in the Alpha F77 compiler that is not available in the Alpha F90 compiler has been implemented in the I64 F90 compiler, including FDML and CDD support. See the Fortran T8.0 product release notes for details.
- Alpha values for the /ARCH and /TUNE qualifiers are accepted on the compiler invocation command for compile-and-go compatibility. An informational message is displayed saying that they are ignored.

For more information about this release, including installation instructions, read the Fortran T8.0 product release notes. To extract the release notes, set your default to the directory that contains the Fortran PCSI kit and enter one of the following commands:

```
$ PRODUCT EXTRACT RELEASE_NOTES FORTRAN ! For TXT file
$ PRODUCT EXTRACT FILE FORTRAN/SELECT=FORTRAN_RELEASE_NOTES.PS ! For PS file
```

5.19 HP MACRO for OpenVMS

The OpenVMS MACRO compiler compiles Macro-32 source code written for OpenVMS VAX systems (the VAX MACRO assembler) into machine code that runs on OpenVMS Alpha and OpenVMS I64 systems. The following notes pertain to the MACRO compiler.

5.19.1 HP MACRO for OpenVMS I64

V8.2

The native MACRO compiler has been provided with OpenVMS I64 Version 8.2. It uses the same DCL commands as the MACRO compiler on OpenVMS Alpha.

Most programs that compile on OpenVMS Alpha should recompile on OpenVMS I64 without modification. However, programs that call routines with nonstandard return values, or programs that use the JSB instruction call routines written in other languages must add some new directives to the source file.

For more information on the new directives and for details on the MACRO compiler, refer to the *HP OpenVMS MACRO Compiler Porting and User's Guide*.

5.19.2 /TIE Qualifier Defaults Differ on Alpha and I64

V8.2

The default for the /TIE qualifier is different on Alpha and I64 systems. On Alpha, the default is /TIE. On I64, the default is /NOTIE.

5.19.3 /OPTIMIZE=VAXREGS Qualifier Not Supported on I64

V8.2

The /OPTIMIZE=VAXREGS qualifier, which is supported on Alpha, is not supported on I64. Unfortunately, all of the related code was not removed from the command line processing. If you specify /OPTIMIZE=ALL on I64, you will accidentally turn on the unsupported VAXREGS optimization. A future release will correct the command line process to avoid turning on the VAXREGS optimization.

5.19.4 CODGENWARN Message Can Be Ignored (Alpha Only)

V8.2

The Macro-32 compiler might generate the following message:

```
%AMAC-W-CODGENWARN, pre-allocation of multiple condition codes overlapped
```

This message can be safely ignored. The generated code is correct. The Macro-32 compiler will be corrected in a future release.

5.19.5 Granularity Support (I64 Only)

V8.2

The Macro-32 compiler does not fully support the /GRANULARITY qualifier or the .PRESERVE GRANULARITY option. In most cases, the compiler generates byte granular code by default, but there are cases where it does not (for example, when accessing an unaligned quadword). This problem will be corrected in a future release.

5.19.6 Integer Divide-by-Zero Error Not Raised By Default (I64 Only)

V8.2

The Macro-32 compiler does not generate the code to detect integer divide-by-zero properly. For dividing by a run-time value, the compiler attempts to generate checking code by using the /ENABLE=OVERFLOW qualifier. However, it checks the wrong operand. It accidentally raises an error when it divides 0 by another number. In a future release, the compiler will be fixed to generate checking

code by default to detect division by zero. That behavior will match the current behavior on OpenVMS VAX and OpenVMS Alpha systems.

5.19.7 Integer Divide By Most Negative Number Crashes Compiler (I64 Only)

V8.2

The following instruction causes the Macro-32 compiler to crash:

```
DIVL3 R0, #^X80000000, R0
BVS 1$
```

This problem will be fixed in a future release

5.19.8 Integer Divide Might Not Set "V" Condition Code Properly (I64 Only)

V8.2

The integer division instructions (DIVL, DIVW, and DIVB) might not set the overflow (''V'') condition code properly. This problem will be fixed in a future release.

5.19.9 Floating Divide-by-Zero Error Not Raised (I64 Only)

V8.2

The Macro-32 floating-point support routines do not detect floating division by zero. The support routines convert the VAX floating values to IEEE floating values and perform the division. Without the check, the division produces an IEEE NaN value. The support routines then attempt to convert the NaN value back into a VAX floating value. That operation raises a floating invalid error. A future release will fix the support routines to properly raise the floating divide-by-zero error.

5.19.10 INSV Instruction Overwrites Extra Memory (I64 Only)

V8.2

The code generated for the INSV instruction overwrites extra memory when the size argument is a literal 32 and the position argument is larger than 0. In those cases, the INSV instruction overwrites too much of the second longword (in the case of a memory destination) or the second register (in the case of a register destination).

In the case where the destination is a memory reference, a workaround is to place a literal 32 into a scratch register and then use that scratch register as the size operand of the INSV instruction.

For example, consider the following code:

```
INSV R2, #15, #32, (R4)
```

This code correctly updates the upper portion of the longword pointed to by R4, but incorrectly writes all of the next longword instead of just the lower portion of the next longword. The program works correctly if you change the code to the following:

```
MOVL #32, Rtmp          ; Where Rtmp is an available register
INSV R2, #15, Rtmp, (R4)
```

In the case where the destination is a register reference, the workaround is to break up the INSV into multiple instructions. For example, change this code:

```
INSV R2, #15, #32, R4
```

to this code:

```
INSV R2, #15, #17, R4   ; Place bottom 17 bits of R2 into top of R4
ASHL #-15, R2, Rtmp     ; Move high part of R2 into temp register
INSV Rtmp, #0, #15, R5  ; Move high 15 bits of R2 into bottom of R5
```

These problems will be fixed in a future release.

5.20 Hypersort Utility

The following notes pertain to Hypersort V08-006 for OpenVMS Alpha and OpenVMS I64 Version 8.2.

As always, continue to use SORT32 as a workaround for any unresolved problems with Hypersort or any functionality not implemented in Hypersort. Release notes for SORT32 are in Section 5.36.

5.20.1 Reporting a Problem to HP

V8.2

If you find a problem with SORT or MERGE, execute the following commands before you submit a problem report:

```
$ WRITE SYS$OUTPUT "WSEXTENT =''F$GETJPI("","WSEXTENT")'"
$ WRITE SYS$OUTPUT "PGFLQUOTA=''F$GETJPI("","PGFLQUOTA")'"
$ SHOW LOGICAL SORTSHR
$ SORT/STATISTICS (or MERGE/STATISTICS)
```

Include the output in your problem report along with the input files that demonstrate the problem.

5.20.2 Large Files Restriction

V8.2

Hypersort V08-010 includes an improved memory allocation algorithm, but Hypersort still sometimes hangs or ACCVIOs with large input files. To reduce the chances of a hang or an ACCVIO, make sure to set page file quota and working set extent as noted in Section 5.20.8. If you encounter a hang or an ACCVIO with Hypersort, use SORT32 instead.

5.20.3 Hypersort and VFC Files Restriction

V7.3-2

Use of VFC files with Hypersort requires /FORMAT=RECORD_SIZE:n.

5.20.4 /FORMAT=RECORD_SIZE Restriction

V7.3-1

Hypersort supports /FORMAT=RECORD_SIZE:n for use with both SORT and MERGE, with the following two restrictions:

- In all cases, if the command-specified RECORD_SIZE is less than the longest record size (LRL) of any record in the input files, the records that are too long are truncated to the RECORD_SIZE size in the sorted output file and the diagnostic message %SORT-E-BAD_LRL is issued. In this situation, the output file should be discarded and the sort should be rerun. The RECORD_SIZE parameter for the SORT command should be revised to a value appropriate to the size of the largest record as shown in the listing of a DIR/FULL command for the input files.

- SORT and MERGE produce output sequential files from input indexed files. The %SORT-E-BAD_LRL diagnostic message can also be issued for this case.

5.20.5 Using Hypersort with Search Lists and Other Uses of Logical Names

V7.3-1

Hypersort does not fully support search lists and logical names used for input files and work files. If you encounter this problem, use SORT32.

5.20.6 Lack of Free Space for Work Files

V7.3-1

Hypersort does not properly terminate if free space is exhausted in all available sort work files. To avoid this problem, do one of the following:

- Allocate sufficient free space for the devices used for sort work files.
- Use SORT32 to detect that work file space has been exhausted.

5.20.7 Input Asterisk (*) Restriction

V7.3

Hypersort does not support asterisk (*) as an input file specification.

5.20.8 Optimal Working Set Extent and Page File Quota Settings

V7.3-1

SORT32 and Hypersort use different sorting and work file algorithms. The relative speed of these utilities depends on the input file and the memory/disk/CPU configuration. Make sure that the working set extent is no more than one third of the page file quota. Both SORT32 and Hypersort perform best with a working set extent appropriate to a specific input file.

5.21 Intel® Assembler (I64 Only)

V8.2

All modules written in Intel assembly language must contain appropriate unwind directives, either by automatic generation using the -Xunwind flag or by explicitly placing unwind directives into the source module. Without accurate unwind information, the operating system's condition handling and exception dispatching does not work, and your program fails in unexpected ways. Programs without accurate unwind information are not supported by OpenVMS. This is a permanent requirement.

5.22 Librarian Utility

The following release notes pertain to the Librarian Utility and Library Service routines.

5.22.1 Linking Against Data-Reduced ELF Object Libraries Not Recommended (I64 Only)

V8.2

DCX data-reduced ELF object libraries are not stored in contiguous space in the library. As a result, the module cannot be directly mapped into process P2 space because data expansion must first occur. The LBR$MAP_MODULE library

service expands and copies the module into process P2 space. This action causes the resulting pages in P2 space to be counted against process quota.

The LBR$UNMAP_MODULE library service recovers those pages, but the pages remain in a heap storage free list and continue to be counted against process quota. For this reason, HP strongly recommends that DCX data-reduced ELF object libraries first be expanded before performing Linker operations.

5.22.2 Failure to Insert or Replace .STB files in an I64 Library (I64 Only)

V8.2

On OpenVMS VAX and Alpha, .STB files can be stored in object libraries. On OpenVMS I64, the Librarian does not do this. For example:

```
$ LIBR/CREATE OBJ$:SOME_LIBRARY OBJ$:SOME_FILE.STB
Librarian T01-23
%LIBRAR-E-NOTELFFILE, TPSSWRKD$:[TPSS.TRACE.IA64.LPIOBJ]TRACEMSG.STB;1
 is not an ELF object or image file
```

Because .STB files are neither objects nor images, the Librarian does not currently put them into an .OLB file on I64 systems.

On Alpha and VAX, however, STB files are formatted as object files. On VAX, .STB files can be used as input to the Linker. On Alpha, .STB file formats are identical to .OBJ file format (that is, nothing distinguishes them except for the .STB file extension). As such, the Alpha Librarian accepts the file, but it cannot be used as input to the Linker.

On I64, the file type (ET_VMS_LINK_STB) is embedded in the .STB file, allowing the Librarian and Linker to determine that the .STB file cannot be processed.

5.22.3 Librarian Fails to Report Errors When Process Quota Too Low

V8.2

The OpenVMS Alpha and I64 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, increase your PGFLQUOTA before running compression, data reduction, or data expansion operations. In addition, ensure that your command procedures check the return status from the LIBRARY command.

5.22.4 Object Module Name Length Problem (I64 Only)

V8.2

The Librarian incorrectly restricts the module name length to 31 characters for I64 (ELF) object and shareable image libraries. This problem will be corrected in a future release. For conventions on specifying image names, see the *HP OpenVMS Version 8.2 New Features and Documentation Overview*.

5.23 Lightweight Directory Access Protocol (LDAP) API

The following sections contain release notes pertaining to the LDAP API.

5.23.1 The Routine ldap_get_option Returns Error -1 When ld Is NULL

V7.3

Using a value of NULL for the `ld` parameter in a call to `ldap_get_options()` results in an error of -1 being returned, rather than the routine returning a set of global default data.

5.23.2 The Routine ber_flatten() Fails to Detect Mismatched Braces

V7.3

The routine `ber_flatten()` does not correctly detect the situation where '{' and '}' format modifiers in a BerElement are incorrectly matched.

5.24 Linker Utility for OpenVMS Alpha

The following release notes pertain to the Alpha Linker Utility.

For I64 Linker release notes, see Section 5.25.

5.24.1 LINK/NATIVE_ONLY Help Text Clarification

V8.2

Online help for LINK/NATIVE_ONLY could be more clearly stated as follows for both Alpha and I64:

> ''Prevents the linker from generating procedure signature information. Procedure signatures are required to allow the native code being linked to interoperate with images translated from either VAX or Alpha binary code. To build an executable or shareable image that calls or can be called by translated code, link it using /NONATIVE_ONLY. Code that is to interoperate with translated images must also be compiled using the /TIE qualifier. (See the associated compiler documentation for details.) ''

5.24.2 Linker Appears to Hang When Many Files Are Specified

V7.3-2

When the RMS_RELATED_CONTEXT linker option is on (the default is RMS_RELATED_CONTEXT=YES) and a nonexistent file is specified in a list of files for the LINK command, the linker's call to LIB$FIND_FILE takes a long time to complete and the linker may appear to hang. Depending on the number of files being linked and the use of logical names in their specification, the linker may take hours to finish because LIB$FIND_FILE locates every combination of the missing file's prefix before displaying a "file not found" message. Note that you cannot terminate the linker process with Ctrl/Y after the linker has called LIB$FIND_FILE.

Workaround

Specify /OPTION in the LINK command. When you press Return, the linker waits for you to enter information into an options file. When you are finished, press Ctrl/Z. To avoid the problem described in this release note, include the following items in the options file:

- On the first line, specify this option:

RMS_RELATED_CONTEXT=NO

With the RMS_RELATED_CONTEXT option set to NO, any missing file listed in this options file will generate an immediate "file not found" message.

- On subsequent lines, specify the files to be linked, using full file specifications (in the form *disk*:[*dir*]*filename.ext*) for *every* file. Full file specifications are required because when you specify RMS_RELATED_CONTEXT=NO, file name ''stickiness'' is disabled.

For example, consider the following LINK command:

```
$ LINK DSK:[TEST]A.OBJ, B.OBJ
```

If you want to specify this command with RMS_RELATED_CONTEXT=NO, you would specify /OPTION and then enter full file specifications for the files to be linked, as follows:

```
$  LINK SYS$INPUT:/OPTION
RMS_RELATED_CONTEXT=NO
DSK:[TEST]A.OBJ, DSK:[TEST]B.OBJ Ctrl/Z
$
```

For more information about the RMS_RELATED_CONTEXT option, refer to the *HP OpenVMS Linker Utility Manual*.

Example

The following example shows how the linker appears to hang when file DOES_NOT_EXIST.OBJ is included in the list and the RMS_RELATED_CONTEXT option is not specified (and therefore defaults to YES).

```
$  DEFINE DSKD$ WORK4:[TEST.LINKER.OBJ.]
$  DEFINE RESD$ ROOT$, ROOT2$, ROOT3$, ROOT4$, ROOT5$, DISK_READ$:[SYS.] ❶
$  DEFINE ROOT$ WORK4:[TEST.PUBLIC.TEST]
$  DEFINE ROOT2$ WORK4:[TEST.LINKER.]
$  DEFINE ROOT3$ WORK4:[TEST.UTIL32.]
$  DEFINE ROOT4$ WORK4:[TEST.PUBLIC.]
$  DEFINE ROOT5$ WORK4:[TEST.PUBLIC.TMP]
$  LINK/MAP/FULL/CROSS/EXE=ALPHA.EXE  RESD$:[TMPOBJ] A.OBJ,-
_$  RESD$:[SRC]B.OBJ,C,DSKD$:[OBJ]D.OBJ,E,RESD$:[TMPSRC]F.OBJ,-
_$  RESD$:[TEST]G.OBJ,RESD$:[SRC.OBJ]H,RESD$:[COM]DOES_NOT_EXIST.OBJ
Ctrl/T NODE6::_FTA183: 15:49:46 LINK CPU=00:02:30.04 PF=5154 IO=254510 MEM=134 ❷
Ctrl/T NODE6::_FTA183: 15:49:46 LINK CPU=00:02:30.05 PF=5154 IO=254513 MEM=134
Ctrl/T NODE6::_FTA183: 15:50:02 LINK CPU=00:02:38.27 PF=5154 IO=268246 MEM=134
Ctrl/T NODE6::_FTA183: 15:50:02 LINK CPU=00:02:38.28 PF=5154 IO=268253 MEM=134
Ctrl/T NODE6::_FTA183: 15:50:14 LINK CPU=00:02:44.70 PF=5154 IO=278883 MEM=134
```

❶ This command defines logical names and equivalents.

❷ Each time you press Ctrl/T, the CPU and IO values increase, but the MEM and PF values do not, indicating that LIB$FIND_FILE is being called.

As shown in the following example, using an options file to set RMS_RELATED_CONTEXT to NO causes the link operation to finish immediately when it encounters the missing file.

```
$  DEFINE DSKD$ WORK4:[TEST.LINKER.OBJ.]
$  DEFINE RESD$ ROOT$, ROOT2$, ROOT3$, ROOT4$, ROOT5$, DISK_READ$:[SYS.]
$  DEFINE ROOT$ WORK4:[TEST.PUBLIC.TEST.]
$  DEFINE ROOT2$ WORK4:[TEST.LINKER.]
$  DEFINE ROOT3$ WORK4:[TEST.UTIL32.]
$  DEFINE ROOT4$ WORK4:[TEST.PUBLIC.]
$  DEFINE ROOT5$ WORK4:[TEST.PUBLIC.TMP.]
$  LINK/MAP/FULL/ CROSS /EXE=ALPHA.EXE SYS$INPUT:/OPTION
RMS_RELATED_CONTEXT=NO
```

```
RESD$:[TMPOBJ]A.OBJ,RESD$:[SRC]B.OBJ,RESD$:[SRC]C,DSKD$:[OBJ]D.OBJ
DSKD$:[OBJ]E,RESD$:[TMPSRC]F.OBJ,RESD$:[TEST]G.OBJ
RESD$:[SRC.OBJ]H,RESD$:[COM]DOES_NOT_EXIST.OBJ Ctrl/Z

%LINK-F-OPENIN, error opening DISK_READ$:[SYS.][COM]DOES_NOT_EXIST.OBJ; as input
-RMS-E-FNF, file not found
$
```

5.24.3 Change in Linker Default Behavior with Library Check

V7.3-1

Previously, the linker's check between the library and the shareable image was too sensitive. It compared against the exact date and time, signaling LINK-I-DATMISMCH, if no match was found. Now, however, it makes only the same check that the image activator does: that is, it uses the GSMATCH criteria to verify compatibility.

The old behavior (check for date and time) can be obtained by setting the logical name LINK$SHR_DATE_CHECK.

5.24.4 Limit of 25 Elements on Stack

Permanent Restriction

Developers who are creating object files should be aware that the linker's internal stack is guaranteed for only 25 elements. Any calculations must be done within this constraint.

5.25 Linker Utility for OpenVMS I64

V8.2

The following release notes pertain to the I64 Linker Utility.

For Alpha Linker release notes, see Section 5.24.

See Section 5.24.1 for a note that applies to both the Alpha and I64 Linker.

5.25.1 Differences Between the I64 and Alpha Linkers

V8.2

Be sure to read the *HP OpenVMS Version 8.2 New Features and Documentation Overview* for a complete description of the differences between the OpenVMS I64 Linker and the OpenVMS Alpha Linker. The *HP OpenVMS Version 8.2 New Features and Documentation Overview* also contains a description of features that are new with the OpenVMS I64 Linker.

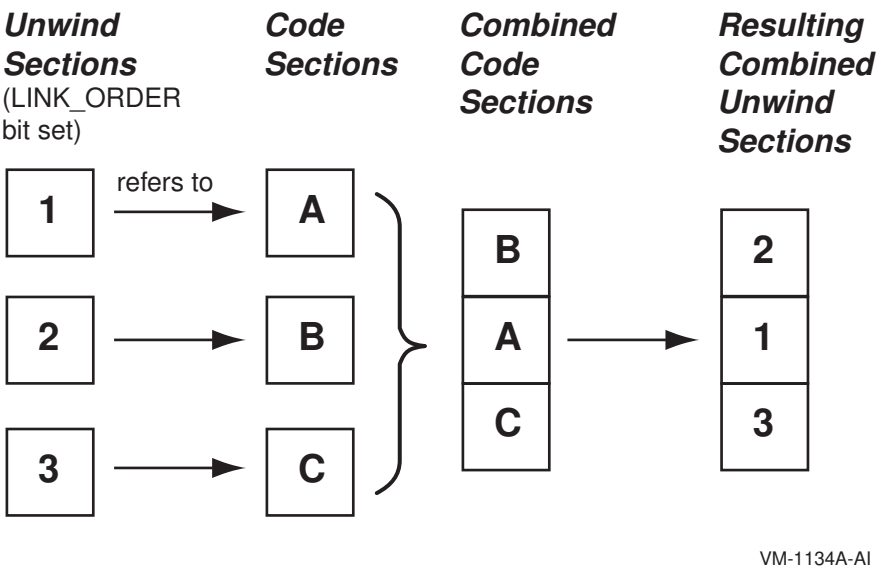
5.25.2 LINK_ORDER Section Header Flag Not Supported

V8.2

The Linker does not support the LINK_ORDER ELF section header flag. However, unwind sections, which have this flag set, are placed in the correct order.

This flag indicates that the section, if combined with other sections in the image file, be combined in the same order as the order of the sections to which they refer are combined. Unwind sections are handled as special cases and appear in the correct order.

The following figure illustrates this concept.



VM-1134A-AI

5.25.3 Linking Against Data-Reduced ELF Object Libraries Not Recommended

V8.2

DCX data-reduced ELF object libraries are not stored in contiguous space in the library. As a result, the module cannot be directly mapped into process P2 space because data expansion must first occur. The LBR$MAP_MODULE library service expands and copies the module into process P2 space. This action causes the resulting pages in P2 space to be counted against process quota.

The LBR$UNMAP_MODULE library service recovers those pages but the pages remain in a heap storage free list and continue to be counted against process quota. For this reason, HP strongly recommends that DCX data-reduced ELF object libraries first be expanded before performing Linker operations.

5.25.4 Errors in the Image Synopsis Section of the Map

V8.2

In the Image Synopsis section of the map, the numeric entities in the fields "Number of code references to shareable images:" and "Estimated map length:" are incorrect. This will be fixed in a future release of OpenVMS I64.

5.25.5 DIFTYPE and RELODIFTYPE Messages

V8.2

On OpenVMS I64 systems, if a module defines a variable as data (OBJECT), it must be referenced as data by all other modules. If a module defines a variable as a procedure (FUNC), it must be referenced as a procedure by all other modules.

When data is referenced as a procedure, the following informational message is displayed:

```
%ILINK-I-DIFTYPE, symbol symbol-name of type OBJECT cannot be
referenced as type FUNC
```

When a procedure is referenced as data, the following informational message is displayed:

```
%ILINK-I-DIFTYPE, symbol symbol-name of type FUNC cannot be
referenced as type OBJECT
```

Type checking is performed by the linker on OpenVMS I64 because the linker must create function descriptors. The equivalent procedure descriptor was created by the compiler on OpenVMS Alpha, so this informational message is new for the linker on OpenVMS I64.

This message is informational only and does not require user action. However, if the linker detects data referenced as a procedure, it might issue the following warning message in addition to the DIFTYPE message:

```
%ILINK-W-RELODIFTYPE, relocation requests the linker to build a
function descriptor for a non-function type of symbol
```

The following example of two modules demonstrates how to fix these errors.

TYPE1.C

```
 #include <stdio>

 int status ;   // Defines status as data.
 extern int sub();

 main ()
 {
     printf ("Hello World\n");
     sub();
 }
```

TYPE2.C

```
 extern int status (int x) ;

 sub ()
 {
 int x;
          x = (int)status;
          return status (x);
 }
```

When these modules are linked, you get an informational message and a warning message, as follows:

```
$ CC/EXTERN_MODEL=STRICT_REFDEF TYPE1
$ CC/EXTERN_MODEL=STRICT_REFDEF TYPE2
$ LINK TYPE1,TYPE2
%ILINK-I-DIFTYPE, symbol STATUS of type OBJECT cannot be referenced as
type FUNC
        module: TYPE2
        file: NODE1$:[SMITH]TYPE2.OBJ;6
%ILINK-W-RELODIFTYPE, relocation requests the linker to build a
function descriptor for a non-function type of symbol
        symbol: STATUS
        relocation section: .rela$CODE$ (section header entry: 18)
        relocation type: RELA$K_R_IA_64_LTOFF_FPTR22
        relocation entry: 0
        module: TYPE2
        file: NODE1$:[SMITH]TYPE2.OBJ;6
```

To fix the problem and avoid the informational and warning messages, correct TYPE1.C to define status as a procedure:

```
TYPE1.C

 #include <stdio>

 int status (int x);  // Defines status as a procedure.
 extern int sub();

 main ()
 {
     printf ("Hello World\n");
     sub();
 }

 int status (int x) {
     return 1;
 }

$ CC/EXTERN_MODEL=STRICT_REFDEF TYPE1
$ CC/EXTERN_MODEL=STRICT_REFDEF TYPE2
$ LINK TYPE1,TYPE2
```

5.26 LTDRIVER: CANCEL SELECTIVE Restriction

Permanent Restriction

In releases prior to OpenVMS Version 6.1, LTDRIVER did not set the "extended DDT" bit; therefore, the POSIX function CANCEL SELECTIVE did not work with LTDRIVER. This problem has been corrected, 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 — such as a LAT connect $QIO) *cannot* be canceled with CANCEL SELECTIVE.

5.27 Mail Utility: Threads Restriction for Callable Mail

V7.1

OpenVMS callable mail routines are *not* thread-safe. Refer to the *Guide to the POSIX Threads Library* for more information about calling non-thread-safe routines within a threaded application.

Because callable mail context information is maintained on a per-process (rather than a per-thread) basis, multiple threads performing context-based processing must be synchronized so that only *one* mail context of a given type is active at once. Otherwise, one thread could corrupt another thread's mail operations.

On OpenVMS Alpha systems, there is an additional restriction when kernel threads is enabled in a multithreaded environment. In this environment, callable mail should be used only in the initial thread.

5.28 OpenVMS Debugger

The following release notes describe conditions specific to the OpenVMS Debugger on OpenVMS Alpha and I64 systems.

For release notes about the Delta and XDelta debuggers, see Section 5.13. For release notes about the System Code Debugger (SCD), see Section 5.37.

Programming Release Notes
5.28 OpenVMS Debugger

5.28.1 General Conditions and Workarounds (I64 Only)

V8.2

The following general conditions have been identified, along with user workarounds:

Condition: Arrays whose elements are on unaligned boundaries are not displayed properly. This occurs when a programming language construct is used to force unnatural data alignment (for example, `#pragma nomember_align` in C or the `PACKED` keyword in Pascal).

Workaround: None.

Condition: When examining an array with scalar (integer) subscripts, the debugger prints the array index values in the currently defined radix, rather than in the decimal radix.

Workaround: Issue the SET RADIX DECIMAL command.

Condition: OpenVMS Debugger on I64 systems displays general, floating-point and predicate registers as if the register rename base (CFM.rrb) and rotating size (CFM.sor) are both zero. In other words, when rotating registers are in use, the effects of the rotation are ignored. This will be corrected in a future release. See Section 5.9 for additional details.

Note

This is a rare condition that only occurs in unusual circumstances in C++ and asssembly language programs; most programs are not affected by this problem.

Workaround: Examine the CFM register and manually adjust the EXAMINE command to account for the nonzero CFM.rrb and CFM.sor fields.

5.28.2 BASIC Language Issues (I64 Only)

V8.2

Condition: When examining a range of a multi-dimensional array, the debugger does not correctly symbolize the element's location. That is, the array row and column values are incorrect.

Workaround: None.

5.28.3 C++ Language Issues (I64 Only)

V8.2

Condition: The SHOW CALLS command sometimes displays C++ mangled names, rather than unmangled names.

Workaround: None.

5.28.4 COBOL Language Issues (I64 Only)

V8.2

Condition: Data declared with EDIT picture clauses are only partially supported. Examining the item (using the EXAMINE command) displays the raw data without scaling or other adjustments. Changing the item by depositing an integer or decimal constant (using the DEPOSIT command) may result in an error.

Workaround: Manually adjust the displayed value by the appropriate scale factor. If you need to change the value, execute the DEPOSIT command using another variable or a quoted string as the source.

5.28.5 Fortran Language Issues (I64 Only)

V8.2

Condition: A deposit of a value into a data item declared as REAL*16 always results in a zero being stored rather than the actual value.

Workaround: None.

5.28.6 Pascal Language Issues (I64 Only)

V8.2

Condition: Arrays declared with dynamic bounds (for example, "array [1..n]") are not always handled correctly. When examining the array, the debugger will sometimes display INVARRDSC or IVALOUTBNDS errors, and SHOW SYMBOL/TYPE may incorrectly display the array bounds.

Workaround: None.

5.28.7 SET SCOPE Command: Behavior Change

V8.2

The behavior of the SET SCOPE command has changed in this release. SET SCOPE now changes the debugger's language setting to the language of the specified scope. Previously, SET SCOPE did not affect the language setting. You can disable this capability by issuing the SET LANGUAGE NODYNAMIC command.

5.28.8 SHOW IMAGE Command Change

V8.2

The SHOW IMAGE command now displays a summary of the address space occupied by the images in your process. To see the full image section information, use SHOW IMAGE/FULL, or specify an image name (for example, SHOW IMAGE FOO).

5.28.9 Client/Server Interface: Previous Versions Not Supported (Alpha Only)

V7.3

The Debugger for OpenVMS Alpha Version 7.3 and later does not support previous versions of the client/server interface. You must install the client/server interface found in the kit on the distribution media, as identified in the following table.

CPU	Operating System	Client Kit
Intel	Microsoft Windows 95, 98, NT, Me, 2000, XP	[DEBUG_CLIENTS011.KIT] DEBUGX86011.EXE
Alpha	Microsoft Windows NT	[DEBUG_CLIENTS011.KIT] DEBUGALPHA011.EXE

These client kits are self-extracting .EXE files.

Once the appropriate executable file has been transferred to the PC, you can run the file to install the debug client on the PC. The InstallShield installation procedure guides you through the installation.

5.29 OpenVMS System Dump Analyzer (SDA)

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

5.29.1 READ Command Default Change

V8.2

The default behavior of the SDA READ command has changed from /LOG to /NOLOG.

5.29.2 CLUE Commands Not Ported to OpenVMS I64

V8.2

The following CLUE commands have not yet been ported to OpenVMS I64 and will return a message to that effect:

```
CLUE CALL
CLUE ERRLOG
CLUE FRU
CLUE REGISTER
```

5.29.3 SHOW CALL_FRAME Functionality Changes (I64 Only)

V8.2

The following functionality changes have been introduced with OpenVMS I64 Version 8.2:

- You can now use the /ALL and /SUMMARY qualifiers with the following commands:
  - SHOW CALL *address*
  - SHOW CALL /HANDLE=*address*
  - SHOW CALL /EXCEPTION_FRAME=*address*
- Previously, the SHOW CALL address always referred to an Invocation Handle address; it can now also be an Exception Frame address.

  If *address* is an Exception Frame address, that frame address is remembered by SDA as a starting point for further call-stack walks.

  If *address* is an Invocation Handle address for a subsequent SHOW CALL command, SDA starts its call-stack walk from the previously remembered starting point.

  If the subsequent *address* is another Exception Frame address, SDA discards the previously remembered call-stack start point and sets the new exception frame address as the current call-stack start point.

Similarly, any SHOW CALL/EXCEPTION_FRAME command also sets the call-stack start point to be the exception frame address given in the command.

This method makes it possible to walk the call stack by starting from an arbitrary exception frame. Note that parts of the call chain can be in process space, even if the starting exception frame is in system space (and vice versa). For the full call-stack walk to work in either case, the SDA current process context must be set to the process where the call stack originates.

- Entering a SHOW CALL command, SHOW CALL/SUMMARY command, or SHOW CALL/ALL command without an address and without a /NEXT, /EXCEPTION_FRAME, or /HANDLE qualifier resets the starting point to the default call chain for the current process.

5.30 PL/I Libraries Not Included in OpenVMS I64 Version 8.2

V8.2

The PL/I libraries (native and translated) are not included in OpenVMS I64 as they are in OpenVMS Alpha. The core PL/I images that are not available in OpenVMS I64 are:

[SYSLIB]DPLI$RTLSHR.EXE
[SYSMSG]PLI$MSG.EXE
[SYSLIB]PLIRTL.IIF
[SYSLIB]PLIRTL_D56_TV.EXE

Applications and shareable images that reference the PL/I library do not work on OpenVMS I64 for Integrity servers. (Typically these are applications and shareable images that contain code written in PL/I.) This restriction applies to both native code and translated VAX and Alpha images.

See a related note in Section 2.8.

5.31 POSIX Threads Library

The following sections contain release notes pertaining to the POSIX Threads Library (formerly named DECthreads).

Also see Section A.8 for a related note.

5.31.1 Stack Overflows During Exception Handling (I64 Only)

V8.2

Exception handling on I64 systems requires considerably more stack space than it does on Alpha. When porting an application from OpenVMS Alpha, if a thread that uses exception handling does not have a generous amount of unused stack space, the thread might experience a stack overflow during exception handling on I64. Usually this appears as an improperly handled ACCVIO that is associated with one of the following operations:

- RAISE
- pthread_cancel
- pthread_exit
- A thread has an active TRY or pthread_cleanup_push block, and an OpenVMS condition is signaled (for example, as a hardware exception or using LIB$SIGNAL or LIB$STOP).

If you see such a problem, try increasing the size of the stack allocated for the thread by a small number of pages. HP recommends initially increasing the stack by 24 KB.

The default stack size has been increased by 24KB to try to address the increased stack usage on I64 systems. If your application creates a large number of threads (using the default size), the application might run out of memory resources. If this happens, you might have to increase process quotas or make application changes to reduce the number of threads that exist simultaneously.

5.31.2 THREADCP Command Behavior on I64 Systems

V8.2

The DCL command THREADCP cannot be used on OpenVMS I64 systems to query and change the two threads-related main image header flags, UPCALLS and MULTIPLE_KERNEL_THREADS. Instead, you must use the DCL commands SET IMAGE and SHOW IMAGE to set and show threads header flags on I64 systems.

Alpha users should continue to use the THREADCP command.

5.31.3 Floating-Point Compilations and Exceptions (I64 Only)

V8.2

Any source modules that call either of the following two old cma threads library routines must be compiled for use on OpenVMS I64 with the /FLOAT=G_FLOAT compiler qualifier (or language-specific equivalent).

```
cma_delay()
```

```
cma_time_get_expiration()
```

These routines accept only VAX-format floating-point values as arguments. Normally, OpenVMS I64 compilers default to using the IEEE formats for floating-point values — unlike OpenVMS Alpha compilers, which default to using VAX formats. These two cma threads routines accept only floating-point arguments in VAX format on both Alpha and I64. Failure to properly compile any calls to these routines may result in unexpected exceptions being raised when IEEE format floating-point values are erroneously passed at runtime.

5.31.4 C Language Compilation Header File Changes

V7.3-2

The following changes have been made in OpenVMS Version 7.3-2:

- INTS.H

  In some prior releases of OpenVMS, the POSIX Threads C language header file PTHREAD_EXCEPTION.H inadvertently contained a #include of the C header file INTS.H. This has been corrected in OpenVMS Version 7.3-2. PTHREAD_EXCEPTION.H no longer causes INTS.H to be included in a compilation. There may be some applications whose compilation requires the presence of INTS.H and which are erroneously relying on PTHREAD_ EXCEPTION.H to provide it.

  Recompiling such application source modules on an OpenVMS Version 7.3-2 system will result in diagnostic messages from the C compiler. These messages identify symbols or data types (for example, int32) that originate in INTS.H and are undefined. To correct such application source modules, add a

direct #include of <ints.h> before the first use of the corresponding symbols or types.

- timespec_t typedef

  In prior releases of OpenVMS, the POSIX Threads C language header file PTHREAD.H contained a definition for a typedef named timespec_t. This is a nonstandard symbol, which does not belong in PTHREAD.H. (This typedef was present for historic reasons related to the contents of C RTL header files such as TIME.H and TIMERS.H.) For OpenVMS Version 7.3-2, the standards compliance of the C RTL and threads header files has been improved. As a result, PTHREAD.H no longer provides the timespec_t typedef.

  There may be some applications whose compilations require the timespec_t typedef, and which erroneously rely on PTHREAD.H to provide it—either directly or indirectly (for example, by using TIS.H). If such an application source module is recompiled on an OpenVMS Version 7.3-2 system, you may get C compiler diagnostic messages listing timespec_t as an unknown symbol or type. To correct such application source modules, either replace the uses of timespec_t with structure timespec, or include the C RTL header file TIMERS.H before the first use of the timespec_t symbol.

  If your application build environment uses a private copy of any older C RTL or threads header files or an extract of them that includes the timespec structure or the timespec_t typedef (both of which are not recommended), you may see an additional compilation error. The compiler may display messages stating that the timespec structure is redefined or defined twice. In such a case, revert to using the system-supplied C RTL and threads header files, or replace the private extracts involving the timespec structure with an inclusion of the system-supplied TIME.H header file.

5.31.5 New Priority Adjustment Algorithm

V7.3-2

As of OpenVMS Version 7.3-2, the adaptive thread scheduling behavior that is described in the *Guide to the POSIX Threads Library* has been implemented with a new priority adjustment algorithm. In some cases, the new algorithm should help avoid problems that can arise when throughput-policy threads of different priorities share synchronization objects. Priority adjustment can also improve application throughput and overall system utilization. Priority adjustment of threads with throughput scheduling policy is automatic and transparent.

5.31.6 Process Dumps

V7.3

If the POSIX Threads Library detects an uncorrectable serious problem at run time (such as data structures that have been damaged by data corruption somewhere in the application), the library may terminate the running image. During termination, the library may trigger creation of a process dump file (which can subsequently be used to diagnose the failure, by way of ANALYZE/PROCESS_DUMP). The size of such a process dump file depends on the size of the process's address space at the time of the failure and can be quite large.

5.31.7 Dynamic CPU Configuration Changes

V7.3

Starting in OpenVMS Version 7.3, the POSIX Threads Library is sensitive to dynamic changes in the number of CPUs that are configured for a running multiprocessor Alpha system. When use of multiple kernel threads is enabled (by way of the LINK/THREADS_ENABLE qualifier or the THREADCP command verb) for an image, the POSIX Threads Library monitors the apparent parallelism of an application and creates multiple kernel threads up to the number of CPUs available. Each kernel thread can be scheduled by the OpenVMS executive to execute on a separate CPU and, therefore, can execute simultaneously.

While an application is running, an operator can stop or start a CPU. Such a dynamic change affects the allowable number of kernel threads that future image activations can create. It also will now affect images that are currently executing.

When a CPU is added or removed, the threads library will query for the new number of active CPUs, and compare this to the number of kernel threads that the process is currently using. If there are now more CPUs than kernel threads, the library will try to spread out the existing POSIX threads over the CPUs (creating new kernel threads as needed, now or in the future). If there are now fewer CPUs than kernel threads, the library will force the extra kernel threads to hibernate, and will reschedule the POSIX threads onto the remaining kernel threads. This will ensure that — so far as the process is concerned — there will not be more kernel threads competing for CPU resources than are available.

5.31.8 Debugger Metering Function Does Not Work

V7.0

The metering capability of the POSIX Threads debugger does not work.

If you use the procedure to debug a running program that is described in Section C.1.1 of the *Guide to the POSIX Threads Library*, your process could fail with an ACCVIO message.

5.32 RMS: Improved Global Buffer VA Management

V8.2

In prior releases, certain patterns of repeated opens and closes of files with RMS global buffers by an application could lead to a VASFUL (virtual address space is full) error and consequently to application failure.

Starting with OpenVMS Version 8.2, RMS has enhanced virtual address (VA) space management, which greatly reduces the possibility of these VASFUL errors.

5.33 RMS Journaling

The following release notes pertain to RMS Journaling for OpenVMS.

For more information about RMS Journaling, refer to the *RMS Journaling for OpenVMS Manual*. You can access this manual on the OpenVMS Documentation CD-ROM (in the archived manuals directory).

5.33.1 Recovery Unit Journaling Incompatible with Kernel Threads

V7.3

Because DECdtm Services is not supported in a multiple kernel threads environment and RMS recovery unit journaling relies on DECdtm Services, RMS recovery unit journaling is not supported in a process with multiple kernel threads enabled.

5.33.2 Modified Journal File Creation

V7.2

Prior to Version 7.2, recovery unit (RU) journals were created temporarily in the [SYSJNL] directory on the same volume as the file that was being journaled. The file name for the recovery unit journal had the form RMS$*process_id* (where *process_id* is the hexadecimal representation of the process ID) and a file type of RMS$JOURNAL.

The following changes have been introduced to RU journal file creation in OpenVMS Version 7.2:

- The files are created in node-specific subdirectories of the [SYSJNL] directory.
- The file name for the recovery unit journal has been shortened to the form: *YYYYYYYY*, where *YYYYYYYY* is the hexadecimal representation of the process ID in reverse order.

These changes reduce the directory overhead associated with journal file creation and deletion.

The following example shows both the previous and current versions of journal file creation:

```
Previous versions:  [SYSJNL]RMS$214003BC.RMS$JOURNAL;1
Current version:  [SYSJNL.NODE1]CB300412.;1
```

If RMS does not find either the [SYSJNL] directory or the node-specific directory, RMS creates them automatically.

5.33.3 Remote Access of Recovery Unit Journaled Files in an OSI Environment

V6.1

OSI 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. It must also be sufficiently unique for the remote node to use this node name to establish a DECnet connection to the host node. This restriction applies only to recovery unit journaled files accessed across the network in an OSI or mixed OSI and non-OSI environment.

5.33.4 After-Image (AI) Journaling

V6.0

You can use after-image (AI) journaling to recover a data file that becomes unusable or inaccessible. AI recovery uses the AI journal file to roll forward a backup copy of the data file to produce a new copy of the data file at the point of failure.

In the case of either a process deletion or system failure, an update can be written to the AI journal file, but not make it to the data file. If only AI journaling is in use, the data file and journal are not automatically made consistent. If additional updates are made to the data file and are recorded in the AI journal, a subsequent roll forward operation could produce an inconsistent data file.

If you use Recovery Unit (RU) journaling with AI journaling, the automatic transaction recovery restores consistency between the AI journal and the data file.

Under some circumstances, an application that uses only AI journaling can take proactive measures to guard against data inconsistencies after process deletions or system failures. For example, a manual roll forward of AI-journaled files ensures consistency after a system failure involving either an unshared AI application (single accessor) or a shared AI application executing on a standalone system.

However, in a shared AI application, there may be nothing to prevent further operations from being executed against a data file that is out of synchronization with the AI journal file after a process deletion or system failure in a cluster. Under these circumstances, consistency among the data files and the AI journal file can be provided by using a combination of AI and RU journaling.

5.33.5 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.

5.34 RTL Library (LIB$)

The following notes pertain to the LIB$ run-time library.

5.34.1 RTL Library (LIB$) Help Omission

V8.2

The OpenVMS Version 8.2 help file for the LIB$ run-time library is missing help for the LIB$LOCK_IMAGE routine. The help file will be corrected in a future release. Meanwhile, refer to the *HP OpenVMS RTL Library (LIB$) Manual* for a complete description of this routine.

5.34.2 RTL Library (LIB$): Calling Standard Routines (I64 Only)

V8.2

This release note clarifies how rotating registers are handled by the following calling standard routines:

LIB$I64_GET_FR
LIB$I64_SET_FR
LIB$I64_GET_GR
LIB$I64_SET_GR
LIB$I64_PUT_INVO_REGISTERS

The Calling Standard invocation context block (ICB) and mechanism vector always record general, floating-point, and predicate registers as if the register rename base (CFM.rrb) and rotating size (CFM.sor) were both zero. In other words, when rotating registers are in use, the effects of the rotation are ignored. This is also true of the register masks used by the LIB$I64_PUT_INVO_REGISTERS routine, because these masks are defined in terms of fields in the ICB structure.

Currently, the supplemental access routines LIB$I64_GET_FR, LIB$I64_SET_FR, LIB$I64_GET_GR, and LIB$I64_SET_GR improperly interpret their register number parameters without applying an adjustment for the effects of the register rename base and rotating size registers. This is an error and will be corrected in a future release.

In the meantime, any code that examines the general, floating-point and predicate registers in an ICB or mechanism vector and that seeks to interpret the register contents as it would be seen by the code during its execution must examine the saved CFM register and apply the appropriate adjustments itself.

5.35 Screen Management (SMG$) Documentation

The following information should be added to topics in the reference section at the end of the *OpenVMS RTL Screen Management (SMG$) Manual*:

V7.2

- The following statement should be added to the Condition Values Returned section of routine SMG$DELETE_VIRTUAL_DISPLAY:

  "Any condition value returned by the $DELPRC system service."

- The description of routine SMG$GET_TERM_DATA contains an error in the Arguments section for the **capability-data** argument. The correction is as follows:

access:	write-only
mechanism:	by reference, array reference

- The description of routine SMG$SET_OUT_OF_BAND_ASTS contains an error in the Arguments section for the **AST-argument** argument. The symbolic names in the Data Structure diagram are incorrect. The symbolic names in the paragraph under this diagram are correct. The correct and incorrect symbolic names are as follows:

Incorrect	Correct
SMG$L_PASTEBOARD_ID	SMG$L_PBD_ID
SMG$L_ARG	SMG$L_USER_ARG
SMG$B_CHARACTER	SMG$B_CHAR

V7.1

- In the documentation for the SMG$READ_COMPOSED_LINE routine, the following text should be appended to the description of the **flags** argument:

  "The terminal characteristic /LINE_EDITING should be set for your terminal for these flags to work as expected. /LINE_EDITING is the default."

- The description of routine SMG$SET_KEYPAD_MODE should contain this note:

  Note

  Changing the keypad mode changes the physical terminal setting. This is a global change for all virtual keyboards, not just the virtual keyboard specified by the **keyboard-id** argument.

5.36 SORT32 Utility

The following release notes pertain to SORT32 V08-010 for OpenVMS Alpha and OpenVMS I64 Version 8.2. For additional notes, refer to Section 5.20.8 and Section 5.20.1.

SORT32 is recommended as a workaround for unresolved problems with Hypersort and for functionality not implemented in Hypersort. Release notes for Hypersort are in Section 5.20.

5.36.1 CONVERT Problem With DFS-Served Disks

V8.2

SORT, MERGE, and CONVERT operations with output directed to a UNIX-served, DFS-mounted disk return a %SORT-E-BAD_LRL error.

To work around this restriction, do one of the following:

- Write the output file to an OpenVMS disk, and then copy the output file to the UNIX-served, DFS-mounted disk.
- Use the CONVERT/FDL command, where the FDL specifies the longest record length (LRL) required for the output file.

5.36.2 Temporary Work Files Not Always Deleted

V7.3-2

SORT32 does not always delete temporary work files. It's a good idea to periodically check SYS$SCRATCH or wherever you put SORT32 work files to see whether any undeleted work files can be deleted to free up disk space.

5.36.3 SORT/SPECIFICATION With Compound Conditions: Requirement

V7.3-1

SORT32 does not issue a diagnostic message for a compound condition in a key specification file unless it is enclosed in parentheses. For example:

Incorrect:

```
/CONDITION=(NAME=TEST1, TEST=(Field2 EQ "X") AND (Field3 EQ "A"))
```

Correct:

```
/CONDITION=(NAME=TEST1, TEST=((Field2 EQ "X") AND (Field3 EQ "A")))
```

5.36.4 Performance Problem with Variable Length Records

V7.3-1

SORT32 allocates fixed-sized slots for sort work files based on the longest record length (LRL) information in the input files. To improve performance, try to set LRL information in the input files as close as possible to the actual longest record length. Poor initial performance might be the result of sorting some files produced by C programs, because the LRL is set higher than necessary (up to 32767).

5.36.5 Work File Directories Restriction

V7.3

SORT32 work files must be redirected to directories that allow multiple file versions that can handle the number of requested work files.

5.37 System Code Debugger (SCD) Not Available on I64 Systems

V8.2

The *HP OpenVMS System Analysis Tools Manual* implies that the System Code Debugger (SCD) is available on both I64 and Alpha systems. In fact, SCD is not available on I64 in Version 8.2. It will be available in a future release.

5.38 System Services

The following notes pertain to OpenVMS system services.

5.38.1 $GETJPI Item Code SCHED_CLASS_NAME Documented Incorrectly

V8.2

In the Version 8.2 *HP OpenVMS System Services Reference Manual* and online help, the $GETJPI item code SCHED_CLASS_NAME is incorrectly documented as CLASS_NAME. This error will be corrected in the next full release.

5.38.2 New GETSYI Item Codes Required for PFN-Mapped Sections (I64 Only)

V8.2

On OpenVMS VAX and Alpha, PFNs are 32-bit numbers (bits 0 to 31). When bit 31 is set, this indicates uncached I/O space.

On OpenVMS I64, PFNs are 64-bit numbers. A program written for I64 that uses PFN-mapped sections must indicate that it has been modified to recognize the larger data size. This is done by setting the new flag SEC$M_ARGS64 in calls to the following system services:

SYS$CRMPSC_PFN_64
SYS$CREATE_GPFN
SYS$CRMPSC_GPFN_64
SYS$MGBLSC_GPFN_64

The SEC$M_ARGS64 flag is ignored on OpenVMS Alpha but must be set on I64. If the flag is not set on I64, the error code SS$_IVSECFLG is returned.

The 32-bit oriented services SYS$CRMPSC and SYS$MGBLSC cannot be used to create or map PFN-mapped sections on I64.

To accommodate the larger data size of PFNs on I64, the SYS$GETSYI system service has been enhanced to recognize a new item code, SYI$_PFN_MEMORY_MAP_64. For details about this item code, refer to the *HP OpenVMS System Services Reference Manual*.

If SYI$_PFN_MEMORY_MAP_64 is specified, the service returns the layout of physical memory into a 64-bit data structure. The new item code and the new layout must be used on I64 systems. OpenVMS Alpha has been enhanced to accept the new form and continues to support the old form.

5.38.3 PFN-Mapped Sections and Uncached Memory (I64 Only)

V8.2

Mapped I/O space on an OpenVMS I64 system might require noncached access. You must set the SEC$M_UNCACHED flag when a PFN-mapped section is created if this section must be treated as uncached memory. The following system services now accept the SEC$M_UNCACHED flag:

SYS$CRMPSC_PFN_64
SYS$CREATE_GPFN
SYS$CRMPSC_GPFN_64

The SYS$MGBLSC_GPFN_64 service accepts but ignores the flag. The CACHED/UNCACHED characteristic is stored as a section attribute, and the system uses this attribute when the section is mapped.

On OpenVMS Alpha systems, all four services accept but ignore the SEC$M_UNCACHED flag.

Note that the older services SYS$CRMPSC and SYS$MGBLSC were not updated and do not accept the new flag.

5.38.4 SYS$ACM: Using on I64 Systems

V8.2

On OpenVMS I64 Version 8.2 systems, the ACME_SERVER process must be started manually. The ACME_SERVER services SYS$ACM requests. If you have applications that use the SYS$ACM system service, add the following commands to your SYS$STARTUP_VMS.COM prodecure before starting applications that use SYS$ACM:

```
$ SET SERVER ACME/START/LOG
$ SET SERVER ACME/CONFIG=(NAME=VMS,CRED=VMS)
$ SET SERVER ACME/ENABLE
```

Note

If you are using LAN Manager authentication supplied by Advanced Server, refer to the Advanced Server V7.3A ECO4 release notes for information about obtaining the necessary I64 components and setup procedure.

The ACME_SERVER process will start automatically on OpenVMS I64 in a future release.

5.38.5 SYS$GOTO_UNWIND Not Available on I64

V8.2

The SYS$GOTO_UNWIND system service is not available on OpenVMS I64 systems; recode to use the new service SYS$GOTO_UNWIND_64.

SYS$GOTO_UNWIND was removed to ensure applications are ported correctly. For more information, refer to *Porting Applications from HP OpenVMS Alpha to HP OpenVMS Industry Standard 64 for Integrity Servers*.

The SYS$GOTO_UNWIND_64 service also exists on OpenVMS Alpha Version 7.3-2 and later, so applications can use common code. For more information about SYS$GOTO_UNWIND_64, refer to online help or to the *HP OpenVMS System Services Reference Manual*.

5.39 Timer Queue Entries (TQEs)

V7.3-1

Management of Timer Queue Entries was redesigned for OpenVMS Alpha Version 7.3-1 to provide significantly higher performance for systems using many TQEs. This change is transparent to nonprivileged applications.

Privileged code can no longer manipulate TQEs directly in any form. In particular, directly accessing pointers in the TQE's queue header (TQEL_TQFL/TQEL_TQBL) causes an access violation in almost all cases. Privileged code may continue to use the internal routines exe_std$instimq/exe$instimq and exe_std$rmvtimq/exe$rmvtimq to enter or remove Timer Queue Entries.

5.40 Traceback Facility

The following notes pertain to the Traceback facility (TRACE).

5.40.1 API Error (I64 Only)

V8.2

This note pertains to the new Traceback Facility API described in the *HP OpenVMS Version 8.2 New Features and Documentation Overview*. Currently, a processing error occurs if the *rel_pc* (relative PC value) or *image_desc* (image name string descriptor) argument is specified as zero. When these arguments are zero, the Traceback Facility should ignore them and continue Traceback processing. Instead, when Traceback encounters these zero values, it discontinues Traceback processing, and returns an unsuccessful return value to the calling program. This problem will be corrected in a future release.

5.40.2 Problem Corrected

V8.2

Previously, the Traceback facility did not correctly handle images installed with the /RESIDENT or /SHARED=ADDRESS_DATA qualifiers. This problem has been corrected. TRACE now displays symbolic locations in images installed with either the /RESIDENT or /SHARED=ADDRESS_DATA qualifiers, provided those images contain symbolic information (that is, they are linked using /TRACEBACK).

5.41 Watchpoint Utility (I64 Only)

V8.2

The Watchpoint Utility has not yet been ported to OpenVMS I64. HP expects to port this utility in a future release.

5.42 Whole Program Floating-Point Mode (I64 Only)

V8.2

On OpenVMS Alpha, the floating-point rounding behavior, exception behavior, and precision control are defined at compile time; each module is independently compiled with its own set of floating-point behaviors. For example, one module can be compiled with a directive that causes overflow calculations to signal an overflow exception, and another module can be compiled with a directive that causes overflow calculations to compute the value InfinityT rather than to signal an exception. When these two modules are combined and run, the code in the modules performs overflow calculations that were specified at compile time.

On OpenVMS I64, the floating-point rounding behavior, exception behavior, and precision control are defined at run time and are guided by the concept of whole program floating-point mode. With whole program floating-point mode, the module that contains the program main entry point (as determined by the Linker) is the module that defines the default floating-point rounding behavior, exception behavior, and precision control.

Most programs are not affected by this difference. Refer to the white paper ''OpenVMS Floating-Point Arithmetic on the Intel® Itanium® Architecture'' for essential reading. You can link to this document from the following web site:

`http://h71000.www7.hp.com/openvms/integrity/resources.html`

6
Hardware Release Notes

This chapter contains information about the following hardware products:

- MP and BMC Console (Section 6.1)
- AlphaServer 2100 (Section 6.2)
- AlphaServer 8200/8400 (Section 6.3)
- AlphaServer ES47/ES80/GS1280 Systems (Section 6.4)
- AlphaServer GS Series (Section 6.5)
- AlphaStation 200/400 (Section 6.6)
- AlphaStation 255 (Section 6.7)
- ATI RADEON 7000 Graphics (Section 6.8)
- ATI RADEON 7500 Graphics (Section 6.9)
- DECwindows X11 Display Server (Section 6.10)
- DIGITAL Modular Computing Components (Section 6.11)
- Digital Personal Workstation (Section 6.12)
- Dual-Controller HSG*nn* device (Section 6.13)
- Open3D Graphics (Section 6.14)
- PowerStorm 300/350 PCI Graphics Controller (Section 6.15)
- RF*nn* DSSI disk devices (Section 6.16)
- RZ*nn* disk devices (Section 6.17)
- ZLX graphics boards (Section 6.18)

A few notes about using device drivers are also included at the end of this appendix.

6.1 MP and BMC Console Restrictions (I64 Only)

The following notes pertain to the MP and BMC consoles.

6.1.1 Input, Output, and Error Device Restriction

V8.2

Currently, the OpenVMS operating system requires that the input, output, and error devices for each MP or BMC console point to a single serial-line console. If your system has an MP console, that is preferable.

If you do not boot from the designated console, a warning is sent to the VMS_LOADER, and you might see other errors during the boot. You might also lose output that you would normally expect to see when booting.

6.1.2 Remapping Ctrl/H to the Delete Key

V8.2

Unlike the OpenVMS operating system, which uses the character 0X7F for DEL/RUBOUT, the MP and BMC consoles and the EFI console environment use Ctrl/H. If you press the delete key on a VT*xxx* terminal, or if you press a key you have mapped to send 0X7F in your terminal emulator, no character is deleted.

Note: The driver does not perform remapping under the following conditions:

- Terminal is in IO$_READALL state.
- Terminal is in IO$_READPBLK state.
- Terminal attribute is set to PASSALL.
- Terminal attribute is set to PASTHRU.
- Pressing Ctrl/V tells the driver to pass the next character and skip the remap check.

6.2 AlphaServer 2100

The following sections contain information specific to the AlphaServer 2100 series computer.

6.2.1 Console Display

V7.2

On AlphaServer 2100 and 2100A systems, a console display similar to the following is normal and does not represent system errors:

```
P00>>>SET CONSOLE SERIAL
P00>>>INIT

VMS PALcode X5.48-112, OSF PALcode X1.35-81

starting console on CPU 0
initialized idle PCB
initializing semaphores
initializing heap
initial heap 1c0c0
memory low limit = 132000
heap = 1c0c0, 13fc0
.
.
.

probing hose 0, PCI
probing PCI-to-EISA bridge, bus 1
probing PCI-to-PCI bridge, bus 2
*** unable to assign PCI base address
*** bus 2, slot 7, function 0, size 00001000 (16 bit I/O)
bus 1, slot 1 -- fra -- DEFEA
bus 1, slot 2 -- vga -- Compaq Qvision
bus 1, slot 3 -- pua -- KFESA
bus 2, slot 1 -- pka -- NCR 53C810
bus 2, slot 6 -- pkb -- NCR 53C810
bus 2, slot 7 -- pkc -- DEC KZPSA
bus 0, slot 7 -- ewa -- DECchip 21041-AA
initializing keyboard

Memory Testing and Configuration Status
Module   Size    Base Addr   Intlv Mode  Intlv Unit  Status
------   -----   ---------   ----------  ----------  ------
  0       64MB   00000000      1-Way         0       Passed
Total Bad Pages 0
```

```
Testing the System
Testing the Disks (read only)
Testing the Network
econfig:            20041 99
econfig:            20042 04
econfig:            20043 00
AlphaServer 2100A Console V4.3-130, built on Oct 26 1996 at 19:44:57
P00>>>P
```

Note that in the previous display, the KZPSA adapter is successfully installed despite the error message displayed in the following lines:

```
*** unable to assign PCI base address
*** bus 2, slot 7, function 0, size 00001000 (16 bit I/O)
```

6.2.2 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 OFFLINE.
```

6.3 AlphaServer 8200/8400: FRU Table Error

V7.2

The error log buffer size is controlled by the system parameter ERLBUFFERPAGES, which has a maximum value of 32 pagelets. If the Field Replaceable Unit (FRU) table exceeds this limit during a boot of the OpenVMS Alpha operating system on an AlphaServer 8200/8400 or 4100 system, an entry will not be written to the error log file.

6.4 AlphaServer ES47/ES80/GS1280 Systems

This section contains release notes of interest to users of AlphaServer ES47/ES80/GS1280 systems. The note in Section 6.5.2 also applies to these systems.

6.4.1 INIT Console Command Usage on ES47/ES80/GS1280 Soft Partitions

V8.2

Usage of the INIT console command on ES47/ES80/GS1280 soft partitions is not supported when other instances within the hard partition are booted and running OpenVMS. Issuing an INIT command might result in system crashes of the other instances running OpenVMS. Shut down the other instances prior to performing an INIT command.

While a console INIT is in progress, do not issue boot commands to other instances within the same hard partition; wait until the INIT has completed.

6.4.2 Setting SYSGEN Parameter PHYSICAL_MEMORY

V7.3-2

Because of hardware configuration requirements on the AlphaServer ES47/ES80/GS1280 systems, HP does not recommend altering the setting of the system parameter PHYSICAL_MEMORY from its default setting of -1. Artificially reducing the amount of memory can produce unpredictable results on these systems.

6.4.3 RAD Support

V7.3-2

OpenVMS support for resource affinity domains (RADs), also known as NUMA support or awareness, has not been qualified in OpenVMS Alpha Version 7.3-2 for the AlphaServer ES47/ES80/GS1280 systems. For more information about RAD support, see the *HP OpenVMS Alpha Partitioning and Galaxy Guide*.

6.4.4 License Requirements

V7.3-2

AlphaServer ES47/ES80/GS1280 systems require a minimum of two OpenVMS software licenses: one license for base support and one license for dual SMP support for the first two processors. This is a change from the previous way of licensing OpenVMS AlphaServer SMP systems. The dual SMP licenses for OpenVMS are included with the CPU modules when you purchase an OpenVMS system or when you purchase additional CPU modules for an OpenVMS system.

6.4.5 STOP/CPU and Shutdown Behavior

V7.3-2

Because of hardware restrictions, any CPU on an AlphaServer ES47/ES80/GS1280 system with an attached I/O drawer cannot be stopped by using the DCL command STOP/CPU. In contrast, CPUs on these systems without an attached I/O drawer can be stopped with this command.

When the shutdown procedure is invoked on an ES47/ES80/GS1280 system with an attached I/O drawer, an error message such as the following might be displayed:

```
%SYSTEM-W-WRONGSTATE, CPU 5 is in the wrong state for the requested operation
```

You can ignore such messages. The shutdown will complete correctly.

6.4.6 Setting Time at MBM

V7.3-2

You must set the correct time and date on the MBM of an AlphaServer ES47/ES80/GS1280 system. If you do not, OpenVMS might display an incorrect time and date.

6.5 AlphaServer GS Series Systems

This section contains release notes of general interest to most users of the AlphaServer GS Series systems.

6.5.1 AlphaServer GS80/160/320 Systems: Device Restriction

Permanent Restriction

Only one set of the following devices found on the legacy bus adapter is configured and supported per partition in OpenVMS Alpha Version 7.3 or higher. These devices include:

- Serial ports COM1 and COM2
- Parallel port
- Keyboard

- Mouse

If multiple legacy bus adapters exist, only the adapter that includes the console port is configured and supported.

6.5.2 OpenVMS Galaxy License Enforcement

V7.3

In an OpenVMS Galaxy computing environment, the OPENVMS-GALAXY license units are checked during system startup and whenever a CPU reassignment between instances occurs.

If you attempt to start a CPU and there are insufficient OPENVMS-GALAXY license units to support it, the CPU will remain in the instance's configured set but it will be stopped. You can subsequently load the appropriate license units and start the stopped CPU while the system is running. This is true of one or more CPUs.

6.5.3 Installing Licenses

V7.3-1

Before you upgrade to Version 7.3-1 or higher, you should perform the following steps to ensure that the common license database can share license units among hard and soft partitions:

1. Calculate required units.
   - Load the base OpenVMS license.
   - Load the SMP licenses.
   - Use the following command to verify that you have the correct number of license units:

     ```
     $ SHOW LICENSE /UNIT_REQUIREMENTS /CLUSTER
     ```

   Note

   The base OpenVMS license allows you to have only one interactive user login per physical system (not per partition). (However, you can always log in from OPA0: in each partition.) For additional interactive users, you will require additional license units. See your HP support representative to determine your needs.

2. Add your licenses to the common license database. For example:

   ```
   $ LICENSE REGISTER license-name /ISSUER=DEC -
   _$ /AUTHORIZATION=USA123456 -
   _$ /PRODUCER=DEC -
   _$ /UNITS=1050 -
   _$ /AVAILABLITY=H -
   _$ /OPTIONS=(NO_SHARE) -
   _$ /CHECKSUM=2-BGON-IAMA-GNOL-AIKO
   ```

   Note that you cannot use the /INCLUDE qualifier with the LICENSE REGISTER command to override the NO_SHARE attribute of the license.

3. Modify the license to override the NO_SHARE attribute of the PAKs with the command LICENSE REGISTER /INCLUDE=(*node-name-list*). For example:

```
$ LICENSE MODIFY OPENVMS-ALPHA /INCLUDE=(NODEA, NODEB, NODEC)
```

4. To make OpenVMS Alpha license units available to the instance of OpenVMS running in each partition, you must ensure that SRM environment variable SYS_SERIAL_NUM is the same in each partition. To do so, perform the following steps:

   a. From the master console of each partition (usually on console line 0), use the SHOW SYS_SERIAL_NUM command to display the system serial number. For example:

```
P00>>>
P00>>>SHOW SYS_SERIAL_NUM
sys_serial_num G2A105
```

   If the value of SYS_SERIAL_NUM is blank, use the SHOW SYS_ SERIAL_NUM command from the master console in each of the other partitions to check for a nonblank system serial number.

Note

If all partition consoles show a blank value for SYS_SERIAL_NUM, you must create a nonzero value of up to 12 characters. Ensure that the system serial number that you create is not used on any other AlphaServer GS80/160/320 on this OpenVMS Cluster.

   b. Once you have determined the system serial number, use the SET SYS_ SERIAL_NUM command from the master console of each partition to change SYS_SERIAL_NUM to the correct value. For example:

```
P00>>>
P00>>>SET SYS_SERIAL_NUM G2A105
```

   You must do this in every hard partition and in every soft partition.

5. In order for the OpenVMS Cluster license database to be updated correctly, HP recommends that you completely shut down and reboot all OpenVMS Cluster common nodes. A rolling upgrade type of boot does not correctly update the common license database.

Note

If your system is part of an OpenVMS Cluster that shares a common license database, anytime you reconfigure the number of hard or soft partitions on your AlphaServer GS80/160/320, you must make sure that all partitions have the same SYS_SERIAL_NUM.

For partitionable machines that are sharing NO_SHARE licenses across partitions, it is possible to see the following error text on system bootup.

```
%LICENSE-E-NOAUTH, DEC OPENVMS-ALPHA use is not authorized on this node
-LICENSE-F-EXCEEDED, attempted usage exceeds active license limits
-LICENSE-I-SYSMGR, please see your system manager
Startup processing continuing...
```

This error text can be safely ignored. The text is displayed when someone has logged into a system that is sharing the OPENVMS-ALPHA PAK and they are then in use. This will be fixed in a future release.

6.5.4 AlphaServer GS60/GS60E/GS140 Multiple I/O Port Module Configuration Restriction

V7.2-1

AlphaServer GS60/GS60E/GS140 configurations with more than a single I/O Port Module, KFTHA-AA or KFTIA-AA, might experience system failures.

When upgrading OpenVMS Galaxy and non-Galaxy AlphaServer 8200/8400 configurations with multiple I/O Port Modules to GS60/GS60E/GS140 systems, customers must install one minimum revision B02 KN7CG-AB EV6 CPU (E2063-DA/DB rev D01) module as described in Compaq Action Blitz # TD 2632.

For complete details about this restriction and its solution, refer to Compaq Action Blitz # TD 2632.

6.6 AlphaStation 200/400: ISA_CONFIG.DAT Changes Required

V7.1

Customers configuring ISA devices on AlphaStation 200/400 Family systems must change their SYS$MANAGER:ISA_CONFIG.DAT file, so that the node information for each device appears at the end of each device description block.

Warning

For upgrades from OpenVMS Version 6.2 or 7.0 systems, this change must be made before starting the upgrade procedure.

Table 6–1 shows the changes to the device description block.

Table 6–1 Changes to Device Description Block

Before Version 7.1	After Version 7.1
[AUA0]	[AUA0]
NAME=AU	NAME=AU
NODE=3	DRIVE=SYS$MSBDRIVER
DRIVER=SYS$MSBDRIVER	IRQ=9
IRQ=9	DMA=(0,1)
DMA=(0,1)	PORT=(388:4,530:8)
PORT=(388:4.530:8)	NODE=3

Customers using SYS$MANAGER:ISA_CONFIG.DAT files should read Section A.7.

6.7 AlphaStation 255: PCI Configuration Restriction

V7.1

The OpenVMS Alpha operating system does not support PCI option cards configured in PCI slot 0 on any AlphaStation 255 series systems.

PCI slot 0 is the lowest physical PCI option slot on AlphaStation 255 series systems. The interrupt signal for this slot is shared with the built-in Ethernet port. Because the OpenVMS Alpha operating system does not currently permit PCI devices to share an interrupt line, a PCI device installed in slot 0 will not function correctly or may cause errors to occur with the built-in Ethernet port. As a result of this restriction, AlphaStation 255 series systems support a maximum of two PCI option cards, configured in slot 1 and slot 2.

6.8 ATI RADEON 7000 Graphics (I64 Only)

V8.2

The following release notes pertain to using the embedded ATI RADEON 7000 graphics adapter on OpenVMS I64 systems.

Note: The resource requirements described in Section 6.9.1 also apply to the embedded ATI RADEON 7000 graphics adapter.

6.8.1 I64 Graphics Support

V8.2

The only graphics option currently supported on OpenVMS I64 systems is the embedded RADEON 7000 PCI adapter.

The ATI RADEON 7500 PCI graphics adapter will be supported on OpenVMS I64 Version 8.2 in the near future. Testing is currently in progress. An announcement will be posted on the following website when support for this graphics card is available:

`http://h71000.www7.hp.com/new/`

6.8.2 Hardware Accelerated 3D Graphics Not Supported on RADEON 7000

V8.2

Hardware accelerated 3D (OpenGL) rendering is not supported on the embedded Radeon 7000 PCI adapter.

6.9 ATI RADEON 7500 Graphics

V7.3-2

The following notes describe resource requirements, enhancements, fixes, and a few restrictions for ATI RADEON 7500 graphics. If you want to consult the HP DECwindows Motif for OpenVMS documentation set, in particular *Managing DECwindows Motif for OpenVMS Systems*, you can link to this document and others from the following website:

`http://www.hp.com/go/openvms/doc`

Note that starting with OpenVMS Version 8.2, the license to use 3D support is included as part of the OpenVMS license. For details, refer to Section 6.14.

6.9.1 Resource Requirements

Support for RADEON graphics requires the following system-wide resources:

- 2 global sections
- 296 KB of global memory

In addition, *each* RADEON card requires the following:

- 3 global sections
- 16 MB plus 1 page of global memory
- 16 MB plus 1 page of buffer object space (32-bit System Space Windows)

The global sections are charged against the GBLSECTIONS system resource, and the 16+ megabytes of global memory are charged against the GBLPAGES and GBLPAGFIL resources.

For example, a single RADEON card requires the following:

- 5 global sections
- 16 MB + 8 KB + 296 KB global memory

These numbers equate to the following values:

GBLSECTIONS	5	
GBLPAGES	33376	(GBLPAGES is in units of 512-byte pagelets.)
GBLPAGFIL	2086	(GBLPAGFIL is in units of 8192-byte Alpha pages.)

A 4-card configuration requires the following:

- 14 global sections
- 296 KB + 4*16 MB + 4*8 KB = 64 MB + 328 KB global memory

These numbers equate to the following values:

GBLSECTIONS	14	
GBLPAGES	131728	(GBLPAGES is in units of 512-byte pagelets.)
GBLPAGFIL	8233	(GBLPAGFIL is in units of 8192-byte Alpha pages.)

6.9.2 New Features in Version 8.2

The following sections describe enhancements that were not included in the VMS732_GRAPHICS-V0200 ECO kit.

6.9.2.1 Improved Performance of Text and Other Bitmap Operations

V8.2

Drawing operations involving text and other bitmap operations in DMA mode (screen depth of 16 or 24) are 10 to 20 times faster than previously, depending on the specific platform and graphics card or bus option (AGP, 33 MHz PCI, 66 MHz PCI).

6.9.2.2 Improved Performance of DMA Operations

V8.2

The performance of DMA drawing operations in general has been improved by a factor of up to four or more. This improvement pertains only to DMA mode (screen depth of 16 or 24).

6.9.2.3 Improved Performance of OpenGL Texture Copies

V8.2

The performance of glCopyTexImage2D has been greatly improved.

6.9.2.4 Screen Resolution 1024x864 Added

A new screen resolution (video mode) of 1024x864 is supported at depths of 8, 16, and 24 bits and refresh rates of 60, 70, 75, and 85 Hz.

6.9.2.5 Hardware-Accelerated Indirect 3D Rendering

The same hardware-specific 3D drawing accelerations that are accessible using LOCAL transport from the client-side, direct-rendering OpenGL/Mesa 3D library (DECW$MESA3DSHR_RADEON) are now available in the DECwindows server to use with network transport (that is, indirect rendering using DECnet or TCP/IP).

The performance of 3D applications that use hardware-accelerated indirect rendering falls between that of local, hardware-accelerated direct rendering and the software-only rendering that was formerly the only option for remote connections.

6.9.3 Problems Fixed (Alpha Only)

V8.2

The following problems have been fixed in Version 8.2:

- OpenGL logic operations (GL_XOR, and so forth) have been implemented for raster drawing operations (glBitmap, glDrawPixels).
- Previously, an OpenGL primitive with an incorrect number of vertices (for example, GL_TRIANGLE_STRIP with only two vertices, GL_QUAD_STRIP with an odd number of vertices or with less than four vertices, or GL_LINES with an odd number of vertices) sometimes caused either a process crash or a server crash, depending on whether the connection was local (that is, direct rendering) or remote.
- Window borders could be corrupted when text was repainted in a window after moving windows on the screen.
- During DECwindows startup, if DRI (direct rendering hardware acceleration) startup failed, the DECwindows server would terminate with an ACCVIO. The server now starts up with DRI disabled on the screens on which DRI startup failed.
- Previously, if QueryBestSize was called with a cursor width and height of zero, the DECwindows server would return a width and height of 0 as the "best" size. One effect of this was that pointer shapes were clipped in the Pointer Options dialog box in the traditional DECwindows Session Manager. Now the server returns a width and height of 64.
- In PIO mode (8-bit screen depth or no Open3D license loaded), the server would occasionally hang, usually in circumstances involving heavy text output.
- The misleading "vmsInitDevice: Wrong device driver loaded" message that the Radeon DDX printed in the error log file when it encountered a PCI Radeon card (instead of an AGP Radeon) has been eliminated. The DDX now prints a message that is clearly informational only, either "Radeon: Found PCI device" or "Radeon: Found AGP device."

- Previously, when a user deleted text by backspacing over it in a DECterm window, parts of some deleted characters were left on the screen.
- A machine check would sometimes occur at boot time because of a race condition between the graphics device driver (SYS$G*DRIVER) and OPDRIVER.

  This problem tended to occur on newer systems (DS25, ES45, ES47, ES80, GS80/160/320/1280) and on older systems when a graphics card used as the console was not on the primary PCI bus.

All of these problems have been fixed.

6.9.4 DECW$OPENGLSHR_RADEON.EXE Renamed to DECW$MESA3DSHR_RADEON.EXE

V8.2

The shareable library SYS$LIBRARY:DECW$OPENGLSHR_RADEON.EXE has been renamed to SYS$LIBRARY:DECW$MESA3DSHR_RADEON.EXE to reflect that this library was built from the Mesa 3D code base. The API and functionality remain the same as in previous releases. The logical name DECW$OPENGLSHR is defined to point to the shareable library using the new file specification.

6.9.5 Support Limitations

V7.3-2

The following functionality is not supported:

- S-Video output
- Digital output
- Dual-head operation

  If you connect monitors to both the DVI port and the analog VGA (CRT) port, you will get identical video output on both screens. This is called **cloned video**. True dual-head operation, with independent video displays on each port, will be supported in a future release.

6.9.6 Video Artifacts at High Refresh Rates

V8.2

At high resolutions (for example, 1920x1440 and 2048x1536) and high refresh rates, and under heavy load conditions, video artifacts might occur on some individual RADEON cards or monitors. If you see such artifacts, try using a lower refresh rate.

6.9.7 OpenGL Supports IEEE Arithmetic Only

V8.2

The OpenGL library supports only IEEE floating point format; VAX floating point is not supported. Use the /FLOAT=IEEE_FLOAT option to compile applications.

6.9.8 DECwindows Server Hangs When Output Is Written to the Graphics Console (OPA0)

V8.2

If output is written to the graphics console (OPA0) after DECwindows has started, the DECwindows server hangs and the screen freezes. Pressing CTRL/F2 allows the DECwindows server to run again.

Instead of writing messages directly to OPA0, HP recommends using OPCOM and the Console Window application to display messages that normally would be written to the console. To enable this application, edit the command procedure SYS$MANAGER:DECW$PRIVATE_APPS_SETUP.COM and add the following global symbol definition:

```
$ DECW$CONSOLE_SELECTION == "WINDOW"
```

If you do not have SYS$MANAGER:DECW$PRIVATE_APPS_SETUP.COM, you can create this command procedure from SYS$MANAGER:DECW$PRIVATE_APPS_SETUP.TEMPLATE.

After editing SYS$MANAGER:DECW$PRIVATE_APPS_SETUP.COM, restart DECwindows with the following command:

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

6.9.9 Monitor Must Be Connected During Initialization

The RADEON 7500 graphics card has two video output ports: one for digital and one for analog. The digital interface is not currently supported. However, using a digital-to-analog adapter, you can plug an analog monitor into the digital port and get the identical output that you get using the analog port. If you use the digital port, the monitor must be attached before the system is powered up in order for the port to function correctly.

6.9.10 Boot Reset Recommendation (Alpha Only)

HP recommends that the console variable `boot_reset` be set to ON.

6.9.11 No Overlay Planes

Hardware overlay planes are not supported.

6.9.12 Single Colormap

The RADEON 7500 graphics controller supports only one hardware colormap. Keep this in mind if you change to the 8-bit color depth, where the default visual is PseudoColor. Attempting to use more than one PseudoColor colormap at a time causes colormap flashing.

Note

3D (OpenGL) rendering is not supported on 8-bit PseudoColor visuals. (See also Section 6.9.18.)

Applications should not install or deinstall colormaps themselves. The window manager should perform these actions. However, the application is responsible for providing the window manager with hints about which colormaps to install or deinstall. You provide this information using the Xlib

function XsetWMColormapWindows( ). This function sets the WM_COLORMAP_ WINDOWS property for a given window.

6.9.13 Single Bit Depth for All Windows

When you are using the RADEON 7500 card, all windows created on a particular head must have the same bit depth. The RADEON 7500 card supports bit depths of 8, 16, and 24 bits per pixel on any graphics head, but once the DECwindows server establishes a bit depth on a particular head, only windows or visuals with that bit depth can be created.

6.9.14 Pixel Depth for Read/Write Color Map

By default, the RADEON 7500 provides a pixel depth of 24 planes with a read-only, TrueColor color map. Some applications, such as DECwindows Paint, require a read/write color map. If Paint is run with a read-only color map, it fails with the following error message:

```
Error: Visual Not Supported
Supported Visuals are {PseudoColor, GrayScale, StaticGray}
```

To use a read/write color map, edit the file SYS$MANAGER:DECW$PRIVATE_ SERVER_SETUP.COM (or, if this file does not exist, create it from SYS$MANAGER:DECW$PRIVATE_SERVER_SETUP.TEMPLATE) and add the following logical name definition to the file:

```
$ DEFINE/EXECUTIVE/SYSTEM/NOLOG DECW$SERVER_PIXEL_DEPTH 8,8,8,8,8,8,8,8
```

Then restart DECwindows using the following command:

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

This change sets the pixel depth to 8 planes (on up to 8 graphics cards, which allows for a multiple-card configuration) and allows the server to provide a PseudoColor visual.

6.9.15 Backing Store/Save Unders Not Supported for 3D Windows

The implementation of backing store and save unders in the RADEON 7500 X11 server does not support 3D windows.

6.9.16 Threads Restriction

The RADEON 7500 OpenGL library for OpenVMS is not thread safe. However, OpenGL can be used in a multithreaded program if the use of OpenGL is restricted to a single thread within the program.

6.9.17 No Support for Single Buffered Visuals

The RADEON 7500 DECwindows server exports only double buffered visuals. For single buffering, users must select a double buffered visual and call `glDrawBuffer( GL_FRONT )` in their application.

6.9.18 No 3D Support for Color Index Mode

Even though 8-bit visuals are supported for 2D operations when the DECwindows server is started with bit depth = 8, OpenGL rendering is not supported on 8-bit visuals.

6.9.19 Timer Mechanism

Under certain circumstances, it is possible for a process to be interrupted while it owns the hardware lock, resulting in an apparent DECwindows server hang.

A timer mechanism has been implemented to detect this situation and to rectify it by forcing an image exit in the suspended process — or, in some instances, by deleting the process.

The timer mechanism can be configured using the following two logicals, which should be specified in the DECW$PRIVATE_SERVER_SETUP.COM file:

- DECW$DRM_TIMER_PERIOD (Default: 5.0 seconds)

  Specifies the duration of the clock tick in seconds; a floating-point value.

- DECW$DRM_TIMEOUT (Default: 6)

  Specifies the number of clock ticks that are allowed to elapse before a timeout occurs and the DECwindows server seizes control of the RADEON card.

The default values are chosen to minimize the impact of the timer on the performance of graphics applications. If you want to reduce the length of time before the DECwindows server begins responding again, you can do so by decreasing the value of DECW$DRM_TIMER_PERIOD.

A process can be interrupted while holding the hardware lock under either of the following conditions:

- The process is remotely logged in with its terminal displayed on a different system.
- The process is a subprocess that has been suspended or terminated by another process in such a way that normal exit handling does not occur.

If neither of these conditions is likely to occur in your configuration, you can disable the timer mechanism entirely by setting the timer period to zero:

```
$ DEFINE/SYSTEM DECW$DRM_TIMER_PERIOD 0
```

Whenever you change the value of DECW$DRM_TIMER_PERIOD, you must either restart the DECwindows server or reboot the system for the changes to take effect. To restart the DECwindows server, use the following command:

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

6.10 DECwindows X11 Display Server (Alpha Only)

This section contains release notes pertaining to the DECwindows X11 display server for OpenVMS Alpha systems.

6.10.1 Vendor and Operating System Version Data Has Been Corrected

V8.2

Numerous instances of vendor and operating system version information have been updated.

6.10.2 S3 Multihead Graphics

Permanent Restriction

Alpha computers equipped with S3 Trio32 or Trio64 graphics cards support single-screen display only. Multihead graphics are not supported.

6.10.3 Integrated Graphics Board Support

V7.3-2

Starting with OpenVMS Version 7.3, support for graphics boards was fully integrated with the OpenVMS operating system. The following new boards are now supported:

- ATI RADEON 7500 (added in 2003)
- Elsa GLoria (PowerStorm 4D10T)
- OXYGEN VX1
- PowerStorm 300
- PowerStorm 350

6.11 DIGITAL Modular Computing Components (DMCC)

This section contains release notes pertaining to DMCC.

6.11.1 Alpha 5/366 and 5/433 PICMG SBC Restriction

Permanent Restriction

The KZPDA SCSI Controller and the PBXGA Graphics Card cannot be placed in a slot behind the bridge on the DIGITAL Modular Computing Components (DMCC) Alpha 5/366 and 5/433 PICMG SBCs.

6.11.2 Updating the SRM Console

Permanent Restriction

To update the SRM console on the Alpha 4/233 (21064a), 4/266 (21164a), 5/366, and 5/433 DMCC systems, you must choose either the SRM console or the AlphaBIOS setup. You can store only one console.

- If you are running OpenVMS on these systems, update only the SRM console.
- If you are running Windows NT on these systems, update only the AlphaBIOS setup.

If you accidentally update both the SRM and the AlphaBIOS consoles, you will enter the AlphaBIOS Setup menu, and you will *not* have the option to return to the SRM console. The only way to exit the AlphaBIOS Setup menu and return to the SRM console is to use a Firmware Update Utility located at the following Internet site:

```
ftp://ftp.digital.com/pub/Digital/Alpha/firmware/index.html
```

6.12 Digital Personal Workstation: Booting OpenVMS V7.3-1 and Higher

V7.3-1

If you are using the Digital Personal Workstation 433au, 500au, and 600au series systems, you can boot OpenVMS Version 7.3-1 or higher from an IDE CD–ROM if the controller chip is a Cypress PCI Peripheral Controller. You cannot boot OpenVMS on a Digital Personal Workstation au series system from an IDE CD–ROM with an Intel Saturn I/O (SIO) 82378 chip in your configuration. You must use a SCSI CD–ROM, if the Intel SIO chip is present.

To determine which IDE chip you have in your configuration, enter the following SRM console command:

```
SHOW CONFIGURATION
```

If you see *Cypress PCI Peripheral Controller*, you can boot OpenVMS.

If you see *Intel SIO 82378*, you will need to use and boot from a SCSI CD–ROM.

6.13 Dual-Controller HSG*nn* with Many LUNs Can Fail Under Heavy I/O Load

V7.3-2

A combination of improvements in driver performance and faster systems has uncovered a limit to the amount of I/O that a dual-controller HSG*nn* storage array configured with a relatively large number of LUNs can handle. When this limit is reached, it is possible for the array to be kept so busy processing I/O that it is unable to complete the normal periodic synchronization between controllers, causing a controller hang or failure and a loss of host access to some or all LUNs until a manual controller reset is performed. In the case of such a controller failure, the Last Failure Codes most likely to be reported on the HSG console are 01960186, 01942088, and 018600A0.

Most HSG*nn* devices will continue to run with no problems. If your site experiences an HSG controller hang or failure when a heavy load is applied and the HSG has more than approximately 24 LUNs configured, you may be able to avoid future hangs or failures by reconfiguring the controller with fewer LUNs or distributing I/O so that the HSG is not so heavily loaded.

This issue is being investigated by the appropriate HP engineering groups.

6.14 Open3D Graphics Licensing Change

V8.2

The 3D graphics display capability has traditionally been licensed separately from the OpenVMS operating system. Since its initial offering, the Open3D layered product has required a separately orderable license. When Open3D software began shipping as part of the OpenVMS operating system, the 3D graphics display feature continued to be a separately licensed capability. An example of this licensing is Open3D licensing to support 3D graphics display with the 3X-PBXGG-AA ATI RADEON 7500 PCI 2D/3D graphics adapter.

Starting with OpenVMS Version 8.2, the 3D graphics display feature is licensed with the operating system for both AlphaServers and Integrity servers. Therefore, the Open3D license is not available for Version 8.2 of OpenVMS. Previous versions of OpenVMS still require the Open3D license to be installed on the system for 3D display operation.

HP will continue to support 3D device drivers shipped with OpenVMS Version 7.3-2 under standard contract or Mature Product Support, depending on your support agreement. Device drivers for the following adapters have shipped with Version 7.3-2 of OpenVMS:

- PowerStorm 300 and 350 PCI graphics adapters (SN-PBXGD)
- ATI RADEON 7500 PCI and AGP graphics adapters (3X-PBXGG)

These adapters will continue to run 3D graphics display under OpenVMS Version 8.2 but will no longer require a license. In addition, the following 2D graphics adapters continue to be supported with OpenVMS Version 8.2:

- ELSA Gloria Synergy (SN-PBXGK)
- 3Dlabs Oxygen VX1 (SN-PBXGF)

The ATI RADEON 7500 PCI graphics adapter will be supported on OpenVMS I64 Version 8.2 in the near future. Testing is currently in progress. An announcement will be posted on the following website when support for this graphics card is available:

```
http://h71000.www7.hp.com/new/
```

6.15 PowerStorm 300/350 PCI Graphics Support for OpenVMS

V8.2

For release notes on the PowerStorm 300/350 PCI graphics controller support for a Compaq Workstation running OpenVMS Alpha, refer to the *PowerStorm 300/350 OpenVMS Graphics Release Notes Version 2.0*. These documents, release notes, and installation guides are shipped with the graphics cards.

Open3D License No Longer Checked

Starting with OpenVMS Version 8.2, the license to use 3D (OpenGL) support is included as part of the OpenVMS license. See Section 6.14 for details.

Defining the DECW$OPENGL_PROTOCOL Logical

When you run a 3D graphics application and display output to a system with a PowerStorm 350/300 graphics card, you must make sure the DECW$OPENGL_ PROTOCOL logical is defined as follows on the system on which you are running the application:

```
$ DEFINE DECW$OPENGL_PROTOCOL DECW$OPENGL_PROTOCOL_V11
```

Problem Fixed

Previously, the P350 would sometimes fail to reinitialize properly on session exit.

This problem has been fixed by two modifications:

- A call to vmsCloseScreen has been added to the device-specific riCloseScreen function (which is called, for example, at CDE session exit), causing the channel to the GB device to be deassigned and allowing the driver to reinitialize the board properly.
- The pixel converter resynchronization algorithm in the device driver has been greatly improved.

6.16 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. The problem can cause data loss, and occurs when reading data from one of these devices, if 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.

6.16 RF*nn* DSSI Disk Devices and Controller Memory Errors

HP 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 6–2, HP recommends that you update the microcode.

The microcode for all models, except 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 both 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 supported (see Table 6–2), and if you have a support contract, contact your HP support representative. Otherwise, contact your authorized reseller.

The earliest supportable revision levels of the DSSI disk microcode are shown in Table 6–2.

Table 6–2 Supported Microcode Revision Levels

Device Type	Minimum Level with Supported Microcode
RF31T	T387E
RF31T+	T387E
RF35	T392D
RF35+	T392D
RF36	V427P
RF73	T392D
RF74	V427P

To display the microcode revision 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. Enter the following commands:

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

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

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

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

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

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 6–3.

Caution

Back up the disk before updating the microcode.

Table 6–3 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 6–3. 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.

6.17 RZ*nn* Disk Drive Considerations

The following notes describe issues related to various RZ disk drives.

6.17.1 RZ25M and RZ26N Disk Drives: Recommendations

V7.1

During the testing of HP supported SCSI disk drives on configurations with DWZZAs and long differential SCSI buses, two drives, RZ25M and RZ26N, were found to have bus phase problems. For this reason, do not use these drives in configurations where the differential bus length connecting DWZZAs equals or exceeds 20 meters.

This recommendation applies only to the RZ25M and RZ26N drives. All other disk drives that are listed as supported in the OpenVMS SPD can be used in configurations to the full bus lengths of the SCSI-2 specification.

6.17.2 RZ26N and RZ28M Disks: Recommended Firmware Support

V6.2-1H3

The minimum firmware revision level recommended for RZ26N and RZ28M disks is Revision 0568.

If the latest firmware revision level is not used with these disks, multiple problems can occur.

6.17.3 RZ26L and RZ28 Disks: Required Firmware for Multihost Use

V6.2

If you install RZ26L or RZ28 disks on a multihost SCSI bus in an OpenVMS Cluster, the disk's minimum firmware revision is 442.

The following sections describe a procedure that you can use to update the firmware on some RZ26L and RZ28 drives. This procedure can only be used with drives that are directly connected to a SCSI adapter on a host system. Drives that are attached through an intelligent controller (such as an HSZ40 or KZPSC) cannot be updated using this procedure. Refer to the intelligent controller's documentation to determine whether an alternative firmware update procedure exists.

Important Note

Only certain RZ26L and RZ28 firmware revisions can be safely upgraded to firmware revision level 442. Refer to Section 6.17.3.1 to determine if your disks are capable of being upgraded to firmware revision level 442. If your disk is capable of supporting firmware revision level 442, use the RZTOOLS utility that is described in Section 6.17.3.2 to update the disk's firmware.

6.17.3.1 Firmware Revision Level 442 Requirements

Only the combinations of disk drives and firmware revision levels listed in Table 6–4 are capable of being upgraded safely to firmware revision level 442. Performing the update procedure on any other combination can permanently damage the disk.

Table 6–4 Revision Level 442 Firmware Compatibility

Disk Drive	Firmware Revision	Disk File Name
RZ26L	440C	RZ26L_442D_DEC.FUP
RZ28	441C or D41C 435 or 436	RZ28_442D_DEC2104.FUP RZ28P4_442C_DEC.FUP

6.17.3.2 Firmware Revision Level 442 Installation Procedure

If you determine that your disk requires revision level 442 firmware and it is capable of being upgraded safely, use the following procedure to update the firmware. (See Table 6–4 for the file name of the disk you are upgrading.)

```
$ RZTOOLS_ALPHA :== $SYS$ETC:RZTOOLS_ALPHA
$ RZTOOLS_ALPHA DKB500 /LOAD=SYS$ETC:filename.FUP
  Read in 262144 bytes.
  Current FW version - X440C
  Upgrading to       - DEC0
  Loading code  ......
  New code has been sent to the drive.
```

6.18 ZLX Graphics Boards Support

V8.2

The following families of graphics controller boards are not supported on OpenVMS Version 8.2:

- ZLX-M Series (PixelVision): ZLX-M1 (PMAGC-AA), ZLX-M2 (PMAGC-BA)
- ZLX-L Series (PixelVision Lite): ZLX-L1 (PMAGC-DA), ZLX-L2 (PMAGC-EA)
- ZLXp-L Series (PixelVision PCI): ZLXp-L1 (PBXGC-A), ZLXp-L2 (PBXGC-B)

As of OpenVMS Version 8.2, only 2D support, using the base 2D capabilities shipped with OpenVMS, is provided for the following families of graphics controller boards. Please do not install Open3D to obtain 2D support for the following:

- ZLX-E Series (FFB): ZLX-E1 (PMAGD-AA), ZLX-E2 (PMAGD-BA), ZLX-E3 (PMAGD-CA)
- ZLXp-E Series (TGA): ZLXp-E1 (PBXGA-A), ZLXp-E2 (PBXGA-B), ZLXp-E3 (PBXGA-C)
- ZLX2-E Series (TGA2): PowerStorm 3D30 (PBXGB-AA), PowerStorm 4D20 (PBXGB-CA)

See Section A.1 for a related note.

6.19 Recompiling and Relinking OpenVMS Device Drivers

The following sections contain release notes pertaining to recompiling and relinking OpenVMS device drivers.

For related release notes, see Section 5.3.

6.19.1 Alpha and VAX SCSI Device Drivers

V7.3-1

All OpenVMS Alpha SCSI device drivers from previous versions of OpenVMS must be recompiled and relinked to run correctly on OpenVMS Version 7.3-1 or higher.

If you have an OpenVMS Alpha SCSI driver that you are upgrading from a version prior to OpenVMS Alpha 7.0, see Section 6.19.2.

Note that for OpenVMS Version 7.1 all OpenVMS VAX SCSI device drivers required recompiling and relinking. OpenVMS VAX device drivers that were recompiled and relinked to run on OpenVMS Version 7.1 will run correctly on OpenVMS Version 7.3 or higher.

6.19.2 OpenVMS Alpha Device Drivers

V7.1

Device drivers that were recompiled and relinked to run on OpenVMS Alpha Version 7.0 do not require source-code changes and do not have to be recompiled and relinked to run on OpenVMS Alpha Version 7.1 and higher. (Note that Alpha SCSI drivers, however, must be recompiled and relinked as described in Section 6.19.1.)

Device drivers from releases prior to OpenVMS Alpha Version 7.0 that were not recompiled and relinked for OpenVMS Alpha Version 7.0 must be recompiled and relinked to run on OpenVMS Alpha Version 7.1 and higher.

OpenVMS Alpha Version 7.0 included significant changes to OpenVMS Alpha privileged interfaces and data structures. As a result of these changes, device drivers from releases prior to OpenVMS Alpha Version 7.0 may also require source-code changes to run correctly on OpenVMS Alpha Version 7.0 and higher. For more details about OpenVMS Alpha Version 7.0 changes that may require source changes to customer-written drivers, refer to the *HP OpenVMS Guide to Upgrading Privileged-Code Applications*.

6.20 Device Driver MON Version Handling

V7.3

As of OpenVMS Version 7.3, when SYSTEM_CHECK is enabled, device driver images with names of the form SYS$nnDRIVER_MON.EXE will be automatically loaded by the system loader. If a corresponding _MON version does not exist, the system will use the default image name: SYS$nnDRIVER.EXE.

6.21 Possible Per-Threads Security Impact on Alpha Device Drivers

V7.2

See Section 5.3.7 for information about how possible per-thread security impacts OpenVMS Alpha device drivers.

6.22 Device IPL Setup for OpenVMS Alpha 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.

6.23 CRCTX Routines Enhanced

V7.1-2

The system routines that you can use to manage the Counted Resource Context Block (CRCTX) data structure have been improved. The following routines now set and check the status (CRCTX$V_ITEM_VALID) of the CRCTX data structure:

- IOC$DEALLOC_CRCTX
- IOC$ALLOC_CNT_RES
- IOC$DEALLOC_CNT_RES
- IOC$LOAD_MAP

These routines have changed as follows:

If you call IOC$DEALLOC_CRCTX with a valid CRCTX status (CRCTX$V_ ITEM_VALID set to 1), the service returns a bad status. If the SYSBOOT parameter SYSTEM_CHECK is set, the system will fail. This prevents users from deallocating a CRCTX when they have valid resources that have not been deallocated.

You must call IOC$ALLOC_CNT_RES with an invalid CRCTX status (CRCTX$V_ ITEM_VALID set to 0). If you call this routine with a valid status, OpenVMS assumes that you will lose the resources mapped by this CRCTX. OpenVMS does not allocate new resources and returns a bad status. If SYSTEM_CHECK is set, the system will fail. IOC$ALLOC_CNT_RES sets the valid bit before it returns.

IOC$DEALLOC_CNT_RES must be called with a valid CRCTX (CRCTX$V_ ITEM_VALID set to 1). If you call IOC$DEALLOC_CNT_RES with an invalid CRCTX, OpenVMS assumes that the other parameters are not valid, and returns a bad status. If SYSTEM_CHECK is set, the system will fail. IOC$DEALLOC_ CNT_RES clears the valid bit before it returns.

IOC$LOAD_MAP must be called with a valid CRCTX. If it is called with an invalid CRCTX (CRCTX$V_ITEM_VALID set to 0), it assumes that the other parameters are also invalid, and returns a bad status. If the SYSBOOT parameter SYSTEM_CHECK is set, the system will fail.

Hardware Release Notes
6.23 CRCTX Routines Enhanced

These improvements indicate to device support and privileged-code application developers whether they need to deallocate scatter gather registers, which are treated by OpenVMS as generic resources. If the CRCTX$V_ITEM_VALID bit is set, IOC$DEALLOC_CNT_RES still needs to be called.

A
Product Retirement Notices

This appendix contains notifications about OpenVMS products that are no longer supported or that are slated for retirement. It also tells you how to find manuals that have been archived.

Freeware

Once a product is retired, HP does not accept or act on problem reports posted against the product. However, for those interested in doing their own development and support, and where contractual and other considerations permit, product installation kits or source code for many former products can be available as OpenVMS Freeware, which is available from the following sources:

- On the Freeware media that ships with the OpenVMS operating system.
- On the World Wide Web at the following address:

  `http://www.hp.com/go/openvms/freeware/`

A.1 Compaq Open3D Layered Product Not Supported in Version 8.2

V8.2

For OpenVMS Version 8.2, the *layered product* Compaq Open3D for OpenVMS Alpha V4.9B and earlier versions is not supported. However, it will continue to be supported on OpenVMS Versions 6.2 and 7.3-2 under Mature Product Support (MPS). This product should not be confused with the Open3D product that is integrated into OpenVMS Version 8.2 and supports new cards such as the PowerStorm 300, PowerStorm 350, and ATI RADEON. The product that is no longer supported is 3D support for PixelVision, FFB, TGA, and TGA2-based graphics in releases following OpenVMS Version 7.3-2. See Section 6.18 for a related release note.

If the unsupported Open3D product is installed on your system, you might encounter problems. For example, the DECwindows display server might hang in a CPU-intensive loop. If Open3D is already installed on your current version of OpenVMS and you plan to upgrade to OpenVMS Version 8.2, you must first ensure that none of the following graphics controller boards are installed on your target system:

- ZLX-M Series (PixelVision): ZLX-M1 (PMAGC-AA), ZLX-M2 (PMAGC-BA)
- ZLX-L Series (PixelVision Lite): ZLX-L1 (PMAGC-DA), ZLX-L2 (PMAGC-EA)
- ZLXp-L Series (PixelVision PCI): ZLXp-L1 (PBXGC-A), ZLXp-L2 (PBXGC-B)

A.2 Open3D Graphics Licensing Change

V8.2

Starting with OpenVMS Version 8.2, the 3D graphics display feature is licensed with the operating system for both AlphaServers and Integrity servers. Therefore, the Open3D license is not available for Version 8.2 of OpenVMS.

For details, refer to Section 6.14.

A.3 DECamds Not Supported on OpenVMS Version 8.2

V8.2

DECamds is not supported on OpenVMS Version 8.2. HP recommends that users transition to using the Availability Manager in place of DECamds. For information about the Availability Manager, refer to the following website:

`http://h71000.www7.hp.com/openvms/products/availman/`

A.4 DECevent Not Supported

V8.2

DECevent and the DIAGNOSE command are not supported on OpenVMS Version 8.2.

Replacements for DECevent include the following:

- **System Event Analyzer (SEA)**

  SEA is now the supported error log analysis tool for OpenVMS and for later hardware platforms (for example, all I64 platforms, and the Alpha DS*nn*, ES*nn*, and most GS*nn* platforms).

  Install the latest available version of WEBES to receive the latest hardware support, including SEA. At a minimum, WEBES Version 4.2 or higher is required on Alpha systems, and Version 4.4 or higher is required on I64 systems.

  For detailed information about operating system requirements and supported hardware for SEA, refer to the WEBES Installation Guide, which is with the other WEBES documentation at the following web site:

  `http://www.hp.com/services/webes`

- **Error Log Viewer (ELV)**

  You can use ELV to quickly examine an error log file that was created on these later hardware platforms. ELV is integrated into the OpenVMS operating system and can be accessed by entering the DCL command ANALYZE/ERROR_LOG/ELV.

  For more information about ELV, refer to the ELV chapter in *HP OpenVMS System Management Utilities Reference Manual*.

A.5 DECwrite Reached End-of-Service Life in December 2004

V8.2

On December 31, 2004, DECwrite reached its End-of-Service life and has been removed from the Software Products Library.

A.6 Error Log Report Formatter (ERF) Unsupported

V8.2

The Error Log Report Formatter (ERF) is no longer supported. If you require this product for use with error logs written on older systems running OpenVMS versions prior to Version 7.2, you can access ERF documentation from the Freeware website:

`http://www.hp.com/go/openvms/freeware/`

The Error Log Viewer (ELV) replaces ERF. For more information, refer to online help for ANALYZE/ERROR_LOG/ELV or see the *HP OpenVMS System Management Utilities Reference Manual*.

Before using ERF, you must convert error log files using ELV or the Binary Error Log Translation utility, which is part of DECevent. DECevent and the DIAGNOSE command are no longer supported, but users who need these tools can download the software and related documentation from the Freeware website:

`http://www.hp.com/go/openvms/freeware/`

Until you install the DECevent software, you will get an error if you try to use the DIAGNOSE command.

A.7 ISA_CONFIG.DAT Unsupported in Future Release

V7.1

Support for using the SYS$MANAGER:ISA_CONFIG.DAT file to configure ISA devices will be discontinued in a future release of OpenVMS Alpha. If you use this file, you should convert to using the ISACFG utility from the console, and the new file-based autoconfiguration method for loading device drivers (as described in *Writing OpenVMS Alpha Device Drivers in C*).

A.8 POSIX 1003.4a Draft 4 Interface May Be Retired

V7.0

The POSIX 1003.4a, Draft 4 (or "d4") interface of the Compaq POSIX Threads Library (formerly named DECthreads) is slated for retirement in a future release. Applications that were written using the POSIX 1003.4a, Draft 4 interface should be migrated to the new POSIX 1003.1c standard (or "pthread") interface provided by the POSIX Threads Library. A compatibility mode for the Draft 4 POSIX 1003.4a interface has been provided in this release to help ease migration. This compatibility mode will be removed in a future release.

A.9 Visual Threads: Not Supported for Version 8.2

V8.2

Visual Threads is not supported on OpenVMS I64 systems or on OpenVMS Alpha Version 8.2. OpenVMS Alpha Version 7.3-2 is the final version on which Visual Threads is supported.

A.10 Last Ordering Dates for Alpha Systems

V8.2

The following deadlines apply for ordering Alpha systems:

Product	Final Order Date	Final Ship Date
Alpha systems	September 30, 2006	December 30, 2006
CPU upgrades	September 30, 2007	December 30, 2007

A.11 Archived Manuals

V8.2

As products are retired and the operating system evolves, certain OpenVMS manuals are archived. Archived manuals are no longer maintained and are not part of the OpenVMS documentation set. However, they are available on the following web site:

`http://www.hp.com/go/openvms/doc`

Click on ‘‘Archived documents’’ in the left sidebar to link to the archived manuals.

The following manuals have been archived with the release of OpenVMS Version 8.2:

VAX MACRO and Instruction Set Reference Manual
OpenVMS VAX RTL Mathematics (MTH$) Manual
OpenVMS VAX System Dump Analyzer Utility Manual

B

Interlocked Memory Instructions (Alpha Only)

The *Alpha Architecture Reference Manual, Third Edition* (AARM) describes strict rules for using interlocked memory instructions. The Alpha 21264 (EV6) processor and all future Alpha processors are more stringent than their predecessors in their requirement that these rules be followed. As a result, code that has worked in the past, despite noncompliance, could fail when executed on systems featuring the 21264 processor and its successors. **Occurrences of these noncompliant code sequences are believed to be rare.** Note that the 21264 processor is not supported on versions prior to OpenVMS Alpha Version 7.1-2.

Noncompliant code can result in a loss of synchronization between processors when interprocessor locks are used, or can result in an infinite loop when an interlocked sequence always fails. Such behavior has occurred in some code sequences in programs compiled on old versions of the BLISS compiler, some versions of the MACRO–32 compiler and the MACRO–64 assembler, and in some HP C and C++ programs.

The affected code sequences use LDx_L/STx_C instructions, either directly in assembly language sources or in code generated by a compiler. Applications most likely to use interlocked instructions are complex, multithreaded applications or device drivers using highly optimized, hand-crafted locking and synchronization techniques.

B.1 Required Code Checks

OpenVMS recommends that code that will run on the 21264 processor be checked for these sequences. Particular attention should be paid to any code that does interprocess locking, multithreading, or interprocessor communication.

The SRM_CHECK tool has been developed to analyze Alpha executables for noncompliant code sequences. The tool detects sequences that may fail, reports any errors, and displays the machine code of the failing sequence.

B.2 Using the Code Analysis Tool (SRM_CHECK)

The SRM_CHECK tool can be found in the following location on the OpenVMS Alpha Version 7.3-2 Operating System CD–ROM:

```
SYS$SYSTEM:SRM_CHECK.EXE
```

To run the SRM_CHECK tool, define it as a foreign command (or use the DCL$PATH mechanism) and invoke it with the name of the image to check. If a problem is found, the machine code is displayed and some image information is printed. The following example illustrates how to use the tool to analyze an image called myimage.exe:

```
$ define DCL$PATH []
$ srm_check myimage.exe
```

The tool supports wildcard searches. Use the following command line to initiate a wildcard search:

```
$ srm_check [*...]* -log
```

Use the -log qualifier to generate a list of images that have been checked. You can use the -output qualifier to write the output to a data file. For example, the following command directs output to a file named CHECK.DAT:

```
$ srm_check 'file' -output check.dat
```

You can use the output from the tool to find the module that generated the sequence by looking in the image's MAP file. The addresses shown correspond directly to the addresses that can be found in the MAP file.

The following example illustrates the output from using the analysis tool on an image named SYSTEM_SYNCHRONIZATION.EXE:

```
** Potential Alpha Architecture Violation(s) found in file...
** Found an unexpected ldq at 00003618
0000360C   AD970130     ldq_l          R12, 0x130(R23)
00003610   4596000A     and            R12, R22, R10
00003614   F5400006     bne            R10, 00003630
00003618   A54B0000     ldq            R10, (R11)
Image Name:    SYSTEM_SYNCHRONIZATION
Image Ident:   X-3
Link Time:      5-NOV-1998 22:55:58.10
Build Ident:   X6P7-SSB-0000
Header Size:   584
Image Section: 0, vbn: 3, va: 0x0, flags: RESIDENT EXE (0x880)
```

The MAP file for system_synchronization.exe contains the following:

```
EXEC$NONPAGED_CODE    00000000 0000B317 0000B318 (      45848.) 2 **  5
SMPROUT               00000000 000047BB 000047BC (      18364.) 2 **  5
SMPINITIAL            000047C0 000061E7 00001A28 (       6696.) 2 **  5
```

The address 360C is in the SMPROUT module, which contains the addresses from 0-47BB. By looking at the machine code output from the module, you can locate the code and use the listing line number to identify the corresponding source code. If SMPROUT had a nonzero base, you would need to subtract the base from the address (360C in this case) to find the relative address in the listing file.

Note that the tool reports *potential* violations in its output. Although SRM_CHECK can normally identify a code section in an image by the section's attributes, it is possible for OpenVMS images to contain data sections with those same attributes. As a result, SRM_CHECK may scan data as if it were code, and occasionally, a block of data may look like a noncompliant code sequence. This circumstance is rare and can be detected by examining the MAP and listing files.

B.3 Noncompliant Code Characteristics

The areas of noncompliance detected by the SRM_CHECK tool can be grouped into the following four categories. Most of these can be fixed by recompiling with new compilers. In rare cases, the source code may need to be modified. See Section B.5 for information about compiler versions.

- Some versions of OpenVMS compilers introduce noncompliant code sequences during an optimization called "loop rotation." This problem can be triggered only in C or C++ programs that use LDx_L/STx_C instructions in assembly language code that is embedded in the C/C++ source using the ASM function,

or in assembly language written in MACRO–32 or MACRO–64. In some cases, a branch was introduced between the LDx_L and STx_C instructions.

This can be addressed by recompiling.

- Some code compiled with very old BLISS, MACRO–32, DEC Pascal, or DEC COBOL compilers may contain noncompliant sequences. Early versions of these compilers contained a code scheduling bug where a load was incorrectly scheduled after a load_locked.

  This can be addressed by recompiling.

- In rare cases, the MACRO–32 compiler may generate a noncompliant code sequence for a BBSSI or BBCCI instruction where there are too few free registers.

  This can be addressed by recompiling.

- Errors may be generated by incorrectly coded MACRO–64 or MACRO–32 and incorrectly coded assembly language embedded in C or C++ source using the ASM function.

  This requires source code changes. The new MACRO–32 compiler flags noncompliant code at compile time.

If the SRM_CHECK tool finds a violation in an image, you should recompile the image with the appropriate compiler (see Section B.5). After recompiling, you should analyze the image again. If violations remain after recompiling, examine the source code to determine why the code scheduling violation exists. Then make the appropriate changes to the source code.

B.4 Coding Requirements

The *Alpha Architecture Reference Manual* describes how an atomic update of data between processors must be formed. The Third Edition, in particular, has much more information on this topic. This edition details the conventions of the interlocked memory sequence.

Exceptions to the following two requirements are the source of all known noncompliant code:

- There cannot be a memory operation (load or store) between the LDx_L (load locked) and STx_C (store conditional) instructions in an interlocked sequence.
- There cannot be a branch taken between an LDx_L and an STx_C instruction. Rather, execution must "fall through" from the LDx_L to the STx_C without taking a branch.

  Any branch whose target is between an LDx_L and matching STx_C creates a noncompliant sequence. For instance, any branch to "label" in the following example would result in noncompliant code, regardless of whether the branch instruction itself was within or outside of the sequence:

```
        LDx_L  Rx, n(Ry)
        ...
label:  ...
        STx_C  Rx, n(Ry)
```

Therefore, the SRM_CHECK tool looks for the following:

- Any memory operation (LDx/STx) between an LDx_L and an STx_C
- Any branch that has a destination between an LDx_L and an STx_C

- STx_C instructions that do not have a preceding LDx_L instruction

  This typically indicates that a backward branch is taken from an LDx_L to the STx_C Note that hardware device drivers that do device mailbox writes are an exception. These drivers use the STx_C to write the mailbox. This condition is found only on early Alpha systems and not on PCI-based systems.

- Excessive instructions between an LDx_L and an STxC

  The AARM recommends that no more than 40 instructions appear between an LDx_l and an STx_C. In theory, more than 40 instructions can cause hardware interrupts to keep the sequence from completing. However, there are no known occurrences of this.

To illustrate, the following are examples of code flagged by SRM_CHECK.

```
** Found an unexpected ldq at 0008291C
00082914   AC300000     ldq_l          R1, (R16)
00082918   2284FFEC     lda            R20, 0xFFEC(R4)
0008291C   A6A20038     ldq            R21, 0x38(R2)
```

In the above example, an LDQ instruction was found after an LDQ_L before the matching STQ_C. The LDQ must be moved out of the sequence, either by recompiling or by source code changes. (See Section B.3.)

```
** Backward branch from 000405B0 to a STx_C sequence at 0004059C
00040598   C3E00003     br             R31, 000405A8
0004059C   47F20400     bis            R31, R18, R0
000405A0   B8100000     stl_c          R0, (R16)
000405A4   F4000003     bne            R0, 000405B4
000405A8   A8300000     ldl_l          R1, (R16)
000405AC   40310DA0     cmple          R1, R17, R0
000405B0   F41FFFFA     bne            R0, 0004059C
```

In the above example, a branch was discovered between the LDL_L and STQ_C. In this case, there is no "fall through" path between the LDx_L and STx_C, which the architecture requires.

Note

This branch backward from the LDx_L to the STx_C is characteristic of the noncompliant code introduced by the "loop rotation" optimization.

The following MACRO–32 source code demonstrates code where there is a "fall through" path, but this case is still noncompliant because of the potential branch and a memory reference in the lock sequence:

```
getlck: evax_ldql  r0, lockdata(r8)  ; Get the lock data
        movl       index, r2         ; and the current index.
        tstl       r0                ; If the lock is zero,
        beql       is_clear          ; skip ahead to store.
        movl       r3, r2            ; Else, set special index.
is_clear:
        incl       r0                ; Increment lock count
        evax_stqc  r0, lockdata(r8)  ; and store it.
        tstl       r0                ; Did store succeed?
        beql       getlck            ; Retry if not.
```

To correct this code, the memory access to read the value of INDEX must first be moved outside the LDQ_L/STQ_C sequence. Next, the branch between the LDQ_L and STQ_C, to the label IS_CLEAR, must be eliminated. In this case, it could be done using a CMOVEQ instruction. The CMOVxx instructions are frequently useful for eliminating branches around simple value moves. The following example shows the corrected code:

```
        movl       index, r2          ; Get the current index
getlck: evax_ldql  r0, lockdata(r8)   ; and then the lock data.
        evax_cmoveq r0, r3, r2        ; If zero, use special index.
        incl       r0                 ; Increment lock count
        evax_stqc  r0, lockdata(r8)   ; and store it.
        tstl       r0                 ; Did write succeed?
        beql       getlck             ; Retry if not.
```

B.5 Compiler Versions

Table B–1 contains information about versions of compilers that might generate noncompliant code sequences and the recommended minimum versions to use when you recompile.

Table B–1 Versions of OpenVMS Compilers

Old Version	Recommended Minimum Version
BLISS V1.1	BLISS V1.3
DEC Ada V3.5	HP Ada V3.5A
DEC C V5.x	DEC C V6.0
DEC C++ V5.x	DEC C++ V6.0
DEC COBOL V2.4, V2.5, V2.6	COBOL V2.8
DEC Pascal V5.0-2	DEC Pascal V5.1-11
MACRO–32 V3.0	V3.1 for OpenVMS Version 7.1-2 V4.1 for OpenVMS Version 7.2
MACRO–64 V1.2	See below.

Current versions of the MACRO–64 assembler might still encounter the loop rotation issue. However, MACRO–64 does not perform code optimization by default, and this problem occurs only when optimization is enabled. If SRM_CHECK indicates a noncompliant sequence in the MACRO–64 code, it should first be recompiled without optimization. If the sequence is still flagged when retested, the source code itself contains a noncompliant sequence that must be corrected.

Alpha computers with 21264 processors require strict adherence to the restrictions for interlocked memory sequences for the LDx_L and STx_C instructions described in the *Alpha Architecture Reference Manual, Third Edition*. To help ensure that uses of interlocked memory instructions conform to the architectural guidelines, additional checking has been added to Version 3.1 of the MACRO–32 Compiler for OpenVMS Alpha.

The *Alpha Architecture Reference Manual, Third Edition* describes the rules for instruction use within interlocked memory sequences in Section 4.2.4. The MACRO–32 for OpenVMS Alpha Version 3.1 compiler observes these rules in the code it generates from MACRO–32 source code. However, the compiler provides EVAX_LQxL and EVAX_STxC built-ins, which allow these instructions to be written directly in source code.

The MACRO–32 Compiler for OpenVMS Alpha Version 4.1 now performs additional code checking and displays warning messages for noncompliant code sequences.

B.6 Recompiling Code with ALONONPAGED_INLINE or LAL_REMOVE_FIRST

Any MACRO–32 code on OpenVMS Alpha that invokes either the ALONONPAGED_INLINE or the LAL_REMOVE_FIRST macro from the SYS$LIBRARY:LIB.MLB macro library must be recompiled on OpenVMS Version 7.2 or higher to obtain a correct version of these macros. The change to these macros corrects a potential synchronization problem that is more likely to be encountered on newer processors, starting with Alpha 21264 (EV6).

Note

Source modules that call the EXE$ALONONPAGED routine (or any of its variants) do *not* need to be recompiled. These modules transparently use the correct version of the routine that is included in this release.

Index

A

Ada compiler
  not yet available on I64, 5–6
After-image journaling, 5–35
Alpha systems
  last order and ship dates, A–4
AlphaServer 2100
  console display, 6–2
  SCSI controller restriction, 6–3
AlphaServer 4100
  FRU table restriction, 6–3
AlphaServer 8200 systems
  FRU table restriction, 6–3
AlphaServer 8400 systems
  FRU table restriction, 6–3
AlphaServer ES47/ES80/GS1280 systems
  INIT console command usage on soft partitions, 6–3
  license requirement, 6–4
  RAD support, 6–4
  setting PHYSICAL_MEMORY system parameter, 6–3
  setting time at MBM, 6–4
  STOP/CPU and shutdown behavior, 6–4
AlphaServer GS Series systems
  device restriction, 6–4
  multiple I/O port restriction, 6–7
AlphaStation 200/400
  ISA_CONFIG.DAT changes required, 6–7
AlphaStation 255
  PCI configuration restriction, 6–8
ANALYZE/DISK/SHADOW command, 4–25
  and dissimilar device shadowing, 4–26
ANALYZE/ERROR_LOG command
  not ported to I64, 3–2
Applications support for current release, 2–1
Archived manuals, A–4
Associated products
  Software Public Rollout Reports, 2–1
  versions supported for current release, 2–1
ATI RADEON 7000 graphics, 6–8
  hardware accelerated 3D graphics not supported, 6–8
ATI RADEON 7500 graphics, 6–8 to 6–14
  DECW$OPENGLSHR_RADEON.EXE renamed, 6–11
ATI RADEON 7500 graphics (cont'd)
  DECwindows server hangs, 6–12
  hardware-accelerated indirect 3D rendering, 6–10
  OpenGL supports IEEE arithmetic only, 6–11
  performance improvements
    DMA operations, 6–9
    OpenGL texture copies, 6–10
    text and other bitmap operations, 6–9
  problems fixed, 6–10
  video artifacts at high refresh rates, 6–11
Authorize utility
  new quotas for DEFAULT and SYSTEM accounts, 4–1
AUTOGEN
  and NEWPARAMS.DAT files, 4–2

B

Backup API
  journaling events, 5–6
Backup Utility
  behavior change, 4–2
BASIC
  V1.5A required to create STARLET library, 2–1
BLISS
  See HP BLISS compiler
BMC console restrictions (I64 only), 6–1
BUGCHECKFATAL system parameter, 5–12

C

C compiler
  See HP C compiler
C programs
  compiling and case sensitivity, 5–6
C RTL, 5–6 to 5–10
  DECC$SHR_EV56 link problem fixed, 5–10
  EABANDONED code in <errno.h>, 5–9
  getc and getchar prefix problem, 5–8
  getc problem, 5–8
  getpwnam_r, getpwuid_r pointer problem, 5–7
  glob issues fixed, 5–10
  iconv prototype change, 5–8
  IPV6 structures packed, 5–9

C RTL (cont'd)
lseek problem, 5–9
Memory leak, 5–7
mktime problem fixed, 5–9
mmap, mprotect changes, 5–7
__PAL_BUGCHK problem fixed, 5–9
<poll.h> problem, 5–9
snprintf overwrite buffer, 5–7
statvfs problem fixed, 5–9
std namespace problem, 5–8
<stddef.h> modified, 5–8
_strtok_r32 and _strtok_r64 now in scope, 5–7
v*scanf and *snprintf namespace problem, 5–8
vsnprintf overwrite buffer, 5–7
C Run-Time Library
See C RTL
C++ compiler
See HP C++ compiler
Calling Standard
See OpenVMS Calling Standard
CANCEL SELECTIVE function, improved use with LTDRIVER, 5–27
CDSA, 5–11
secure delivery ADK, 5–11
Circuit switching
and reduced performance, 4–11
CLUE commands
not ported to I64, 5–30
Cluster compatibility kits, 4–9
Clusters
See OpenVMS Cluster systems
CLUSTER_CONFIG.COM
limits on root directory names, 4–11
CMAP files
new, 2–2
COB$SWITCHES, 5–15
COBOL RTL
See HP COBOL RTL
COM for OpenVMS
error with heavy load of applications, 2–2
support, 2–2
Common Data Security Architecture
See CDSA
Compaq Open3D layered product
not supported in V8.2, A–1
Compilers
noncompliant code, B–1, B–5
CONVERT-I-SEQ error
CONVERT/NOSORT, 5–12
CPUSPINWAIT bugcheck, 5–12
CRCTX routines enhanced, 6–23
CREATE/MAILBOX command
temporary restriction, 3–2
Ctrl/H key sequence
remapping to DEL (I64 only), 6–2

D

Data-reduced ELF object libraries
linking against, 5–25
DCE RPC
See HP DCE RPC for OpenVMS
DCL command procedures
using SET SHADOW and SHOW SHADOW, 4–23
DCL commands, 3–2
DDB structure
updates to, 5–3
Debugger
See OpenVMS Debugger
DEC PL/I, 2–3
DECamds
not supported on V8.2, A–2
DECC$SHR_EV56 link problem fixed, 5–10
DECdfs for OpenVMS
Version 2.4 required, 2–3
DECdtm
Oracle 8*i*, 9*i*, 4–2
DECevent, A–3
not supported, A–2
DECforms Web Connector
running on OpenVMS Version 7.3-1 and higher, 2–3
Decimal Support RTL
See HP Decimal Support RTL
DECmigrate
not on V8.2 Open Source Tools CD-ROM, 3–3
DECnet for OpenVMS, 1–3
DECnet-Plus for OpenVMS, 1–3
new version required, 1–12
DECnet/OSI
See DECnet-Plus for OpenVMS
DECram
See HP DECram
DECRAM
conflict with DECRYPT command, 2–7
DECwindows Motif
See HP DECwindows Motif
DECwindows X11 display server, 6–14 to 6–15
graphics boards support, 6–21
peripheral connection requirements, 1–8
DECwrite
support ending, A–2
Delete key
requires remapping (I64 only), 6–2
Delta/XDelta debuggers, 5–13
DELTA unavailable on I64, 5–13
register display considerations, 5–13
XDELTA platform differences, 5–13
XDELTA restrictions on I64, 5–13

Device drivers
IPL setup, 6–23
MON version handling, 6–22
per-thread security impact, 6–22
recompiling and relinking, 6–22
SCSI, 6–22
DEVICE_NAMING system parameter, 4–3
DEVOFFLINE error
on ANALYZE/DISK/SHADOW command, 4–25
DIAGNOSE command
not supported, 3–2, A–2
DIGITAL Modular Computing Components (DMCC)
KZPDA controller and PBXGA graphics card, 6–15
updating the SRM console, 6–15
Digital Personal Workstation, 6–15
DIRECTORY command
changes to output, 4–15
Dissimilar device shadowing (DDS)
ANALYZE/DISK/SHADOW command behavior, 4–26
KZPDC restriction, 4–24
Smart Array 5300 restriction, 4–24
write bitmap interaction, 4–23
Documentation changes and corrections
archived manuals, A–4
$GETJPI help error, 5–39
Guide to OpenVMS File Applications, 5–14
HP OpenVMS Cluster Systems manual, 4–11
HP OpenVMS System Management Utilities Reference Manual, 4–20
HP OpenVMS System Services Reference Manual, 5–39
LIB$ help omission, 5–36
LINK/NATIVE_ONLY help text clarification, 5–22
online help, 4–20
OpenVMS Performance Management, 4–20
OpenVMS RTL Screen Management (SMG$) Manual, 5–37
DSSI disk devices
microcode revision levels, 6–18
Dual-controller HSG*nn*
failure, 6–16
DUMP_DEV errors (I64 Only), 4–4
Dynamic CPU configuration
POSIX Threads Library, 5–34

E

EABANDONED code in <errno.h>, 5–9
ECP (Enterprise Capacity and Performance), 4–3
EDIT/FDL
fixing recommended bucket size, 4–3
EFI tools
VMS_SHOW DUMP_DEV errors (I64 Only), 4–4
EFI$CP utility
use not recommended, 4–3
ELV
See Error Log Viewer (ELV) utility
Encryption for OpenVMS Alpha, 1–10
Enterprise Capacity and Performance (ECP)
See ECP
Error Log Report Formatter (ERF)
unsupported, A–3
Error Log Viewer (ELV) utility, 4–4
EV6 Alpha processor, B–1
Extended DDT bit
problem corrected, 5–27
Extended File Cache (XFC), 4–22
External authentication, 4–4
I64 support, 4–4
password expiration notification, 4–5
SET PASSWORD command, 4–4

F

Fast lock remastering, 4–6
Fast Path
turning off, for Galaxy on ES40, 4–14
Fibre Channel
compatibility kits, 4–9
multipath failover restriction for tape devices, 4–13
Firmware
for Alpha servers, 1–9
for Integrity servers, 1–6
Floating-point data
considerations for applications, 5–5
FMS kits, 2–4
Fortran
See HP Fortran
Freeware, 3–1, A–1
menu unavailable on OpenVMS I64, 3–1

G

Galaxy
definitions, 4–14
`getc` and `getchar` prefix problem, 5–8
`getc` problem, 5–8
`getpwnam_r`, `getpwuid_r` pointer problem, 5–7
Gigabit Ethernet switch restriction, 4–12
`glob` issues fixed, 5–10
Graphical Configuration Manager (GCM)
supported for Galaxy, 4–14
Graphical Configuration Utility (GCU), 4–14
Graphics
support for I64 systems, 6–8
Graphics boards support, 6–21

H

Hard partition, 4–14
Help
  changes to, 3–2
HP BLISS compiler
  consequences of noncompliant code, B–1
  warnings (I64 only), 5–14
HP C compiler
  consequences of noncompliant code, B–1
HP C++ compiler
  consequences of noncompliant code, B–1
HP COBOL RTL, 5–15
  SET and COB$SWITCHES, 5–15
HP DCE RPC for OpenVMS, 2–4 to 2–6
HP Decimal Support RTL, 5–15
HP DECram, 2–6
  command conflict between DECRAM and DECRYPT, 2–7
  maximum disk size, 2–7
  removal before upgrade (Alpha only), 1–11
  ships as a SIP on V8.2, 2–6
  Version 2.5 (VAX only), 2–6
HP DECwindows Motif
  keyboard support restrictions (I64), 1–8
  language variant availability, 2–8
  pause screen can fail to unlock (Alpha only), 2–8
  startup messages, 1–8
  support for LAT transport interface, 2–8
  user-written transport support, 2–9
  version support, 1–4
HP Fortran
  for I64, 5–16
HP MACRO for OpenVMS, 5–16 to 5–19
  CODGENWARN message (Alpha only), 5–17
  floating divide-by-zero error not raised (I64 only), 5–18
  granularity support (I64 only), 5–17
  INSV instruction overwrites extra memory (I64 only), 5–18
  integer divide by most negative number crashes compiler (I64 only), 5–18
  integer divide might not set ‘‘V’’ condition code properly (I64 only), 5–18
  integer divide-by-zero error not raised by default (I64 only), 5–17
  on I64 systems, 5–17
  /OPTIMIZE=VAXREGS qualifier not supported on I64, 5–17
  /TIE qualifier defaults on Alpha and I64, 5–17
HP Secure Web Browser
  increased memory requirement, 3–3
  installation error on ODS-2 (I64 only), 3–4
HP Secure Web Server
  support, 2–9
HP SSL installation, 1–4
HSG*nn*
  failure, 6–16
Hypersort utility, 5–19 to 5–20

I

iconv
  fixes to, 3–3
`iconv` prototype change, 5–8
IDE CD–ROM, 6–15
INIT console command
  usage on ES47/ES80/GS1280 soft partitions, 6–3
INITIALIZE command
  new /ERASE keywords, 4–5
  new /GPT qualifier, 4–7
  new behavior when /CLUSTER_SIZE not specified, 4–5
INSTALL utility, 4–6
Installation and upgrade information
  networking options, 1–3
Installation error
  HP Secure Web Browser, 3–4
Integrated graphics boards, 6–15
Integrity server
  configurations, 1–5
  firmware, 1–6
Intel® Assembler (I64 only), 5–20
Interlocked memory instructions, B–1
Invocation context block, 5–36
IPL requirement
  VMS fork thread creation, 5–5
IPV6 structures packed, 5–9
ISA_CONFIG.DAT
  unsupported in future release, A–3

K

Kerberos, 1–11
  removing V1.0 before upgrade, 1–11
Kernel threads
  incompatibility with recovery unit journaling, 5–35
KPB extensions, 5–2

L

LANCP
  converting device database after upgrading (Alpha only), 1–12
LDAP API problem, 5–22
LIB$I64_GET_FR, 5–36
LIB$I64_GET_GR, 5–36
LIB$I64_PUT_INVO_REGISTERS, 5–36
LIB$I64_SET_FR, 5–36

LIB$I64_SET_GR, 5–36
LIB$LOCK_IMAGE
  missing from help, 5–36
LIBOTS2
  See HP Decimal Support RTL
Librarian utility, 5–20
  error reporting problem, 5–21
  linking against data-reduced ELF object libraries (I64 restriction), 5–20
  object module name length problem (I64 only), 5–21
  restrictions with .STB files (I64 only), 5–21
LIBRTL
  Calling Standard routines (I64 Only), 5–36
Licensing issues, 6–5 to 6–7
LINK/NATIVE_ONLY help text clarification, 5–22
Linker for OpenVMS Alpha, 5–22 to 5–24
  change in behavior with library check, 5–24
  hangs when processing many files, 5–22
  limit of 25 elements on stack, 5–24
  LINK/NATIVE_ONLY help text clarification, 5–22
  RMS_RELATED_CONTEXT option, 5–22
Linker for OpenVMS I64, 5–24 to 5–27
  data-reduced ELF object libraries, 5–25
  differences from OpenVMS Alpha Linker, 5–24
  DIFTYPE and RELODIFTYPE messages, 5–25
  errors in map, 5–25
  LINK/NATIVE_ONLY help text clarification, 5–22
  LINK_ORDER section header flag, 5–24
Linker map
  errors in image synopsis section, 5–25
LINK_ORDER ELF section header flag, 5–24
Locales
  new, 2–8
Lock Manager
  fast lock remastering, 4–6
  lock value block extended, 4–6
Logical Disk (LD) utility
  problem fixed, 4–7
lseek problem, 5–9
LTDRIVER restriction, 5–27

M

MACRO for OpenVMS
  See HP MACRO for OpenVMS
MACRO–32 compiler
  consequences of noncompliant code, B–1
  recompiling code, B–6
MACRO–64 assembler
  consequences of noncompliant code, B–1
Mail utility (MAIL)
  problem when callable mail used with kernel threads, 5–27
MEDOFL error
  on ANALYZE/DISK/SHADOW command, 4–25
Memory leak, 5–7
Microcode revision levels
  commands for updating, 6–19
  on DSSI disk devices, 6–18
mktime problem fixed, 5–9
mmap, mprotect changes, 5–7
MMG_CTLFLAGS system parameter, 4–20
MP console restrictions (I64 only), 6–1
Multipath failover
  Fibre Channel tape device restriction, 4–13
  tape robots, 4–13
MULTIPROCESSING system parameter, 5–12

N

Network options, 1–3
NEWPARAMS.DAT files
  and AUTOGEN, 4–2
Noncompliant code, B–1, B–2

O

On-Disk Structure
  layout changes, 4–7
Online help
  See Help
Open3D graphics
  controller boards support, 6–21
  licensing change, 6–16
OpenVMS Calling Standard, 5–10
OpenVMS Cluster systems, 4–8 to 4–13
  compatibility kits, 4–9
  compatibility kits for mixed versions, 4–9
  Gigabit Ethernet switch restriction, 4–12
  limits on root directory names, 4–11
  performance reduced with CI-LAN switching, 4–11
  rolling upgrades, 1–9
  SCSI multipath failover, 4–10
OpenVMS Debugger, 5–27
  Alpha
    previous versions not supported, 5–29
  Alpha and I64
    change to SET SCOPE command, 5–29
    change to SHOW IMAGE command, 5–29
  I64
    BASIC language issues, 5–28
    C++ language issues, 5–28
    COBOL language issues, 5–29
    FORTRAN language issues, 5–29
    general conditions and workarounds, 5–28
    Pascal language issues, 5–29
OpenVMS Galaxy, 4–13 to 4–14
  and ES40
    turning off Fast Path, 4–14
    uncompressed dump limitation, 4–14

OpenVMS Galaxy (cont'd)
license enforcement, 6–5
OpenVMS I64
booting from DVD, 1–7
OpenVMS Management Station, 4–15
OpenVMS Registry
Version 2 format database corruption, 4–15
OpenVMS System Dump Analyzer
CLUE commands not ported to I64, 5–30
READ command default changed, 5–30
SHOW CALL_FRAME functionality changes (I64 only), 5–30

P

__PAL_BUGCHK
problem fixed, 5–9
Partition
hard, 4–14
soft, 4–14
Pascal
reinstalling after an upgrade (Alpha), 2–10
V5.8A required to create STARLET library (Alpha only), 2–10
Patch kits
required for mixed-version OpenVMS Cluster system, 4–9
PATHWORKS for OpenVMS (Advanced Server)
displaying ACEs, 4–15
PCB$T_TERMINAL
size increase, 5–3
PCI configuration restriction, 6–8
PE1 system parameter, 4–6
PEdriver
response to LAN congestion, 4–11
Per-thread security
impact on device drivers, 5–4
impact on privileged code, 5–4
PGFLQUOTA problems, 5–21
PL/I
libraries not included in I64, 5–31
RTL support, 2–3
<poll.h> problem, 5–9
POOLCHECK system parameter, 5–12
Port driver $QIO
restriction, 5–27
POSIX Threads Library, 5–31 to 5–34
debugger metering does not work, 5–34
Dynamic CPU configuration, 5–34
floating-point exceptions (I64 only), 5–32
POSIX 1003.4a, Draft 4 interface to be retired, A–3
stack overflows during exception handling (I64 only), 5–31
THREADCP command behavior for I64, 5–32
PowerStorm 300/350 PCI graphics support, 6–17
Open3D license no longer checked, 6–17
Privileged data structures
64-bit logical block number, 5–2
CPU name space, 5–2
forking to a dynamic spinlock, 5–3
KPB extensions, 5–2
PCB$T_TERMINAL size increase, 5–3
per-thread security impact on, 5–4
UCB/DDB updates, 5–3
updates to, 5–1 to 5–4

R

Rebooting on I64 systems, 1–8
Recompiling programs
for Alpha, 5–1
for I64, 5–1
Recovery unit journaling
file creation changes, 5–35
kernel threads, 5–35
remote access of journaled files, 5–35
restriction, 5–36
Remedial kits
obtaining, 1–3
required for mixed-version OpenVMS Cluster system, 4–9
Retired products information, A–1 to A–4
RF73 and RF*nn* disks, controller memory errors, 6–17
RMS Journaling, 5–35
after-image journaling, 5–35
journal file creation modification, 5–35
remote access of recovery unit journal files, 5–36
Rotating registers, 5–36
and ICB, 5–10
and mechanism vector, 5–10
RU journaling
See Recovery unit journaling
RZ*nn* disk drives, 6–20 to 6–21

S

SCD
See System Code Debugger
SCSI controllers
restrictions on AlphaServer 2100 systems, 6–3
SCSI device drivers, 6–22
SCSI multipath incompatibility, 4–10
SDA
See OpenVMS System Dump Analyzer
Security auditing fixed, 4–16
Server Management Process (SMHANDLER), 4–15
SET DEVICE/SWITCH command, 4–13
SET PASSWORD command, 4–4

SET SHADOW
using in DCL command procedures, 4–23
SHADOW_MAX_UNIT
default setting, 1–13
SHAD_MAX_UNIT
memory consumption, 1–13
SHOW LICENSE
/HIERARCHY requires SYSLCK privilege (I64 only), 3–3
/OE requires SYSLCK privilege (I64 only), 3–3
SHOW SHADOW
using in DCL command procedures, 4–23
Smart Array 5300
Volume Shadowing restriction, 4–24
SMG$
documentation corrections, 5–37
SMHANDLER
See Server Management Process
snprintf overwrite buffer, 5–7
Software Public Rollout Reports, 2–1
SORT32 utility, 5–20, 5–38 to 5–39
SPLINVIPL bugcheck, 5–5
SRM_CHECK tool, B–1
SSL installation, 1–4
statvfs problem fixed, 5–9
std namespace problem, 5–8
<stddef.h> modified, 5–8
_strtok_r32 and _strtok_r64 now in scope, 5–7
Support policy for software, 1–1
SYS$ACM
using on I64, 5–40
SYS$STARLET_C.TLB
upgrade error, 1–10
SYSGEN
security auditing fixed, 4–16
SYSMAN
DUMP_PRIORITY, 4–16
System Code Debugger
not available on I64, 5–39
System crashes
recovery from (I64 only), 4–1
System disk
incompatibility with older systems, 1–3
System Event Analyzer (SEA) utility
support on I64, 2–10
System Event Log (SEL)
clearing on Integrity servers, 1–5
System hangs
recovery from (I64 only), 4–1
System parameters, 4–16 to 4–20
BUGCHECKFATAL, 5–12
changes, 4–18
DEVICE_NAMING
used to increase device unit number maximum, 4–3
displaying obsolete, 4–17
MMG_CTLFLAGS documentation error, 4–20
System parameters (cont'd)
MULTIPROCESSING, 5–12
new parameters, 4–17
obsolete parameters, 4–17
PE1, 4–6
PHYSICAL_MEMORY, 6–3
POOLCHECK, 5–12
SYSTEM_CHECK, 5–12
System services
$GETJPI item code SCHED_CLASS_NAME documented incorrectly, 5–39
PFN-mapped sections
and uncached memory (I64 only), 5–40
new item codes required (I64 only), 5–39
SYS$GOTO_UNWIND, 5–41
SYS$GOTO_UNWIND_64, 5–41
using SYS$ACM on I64 systems, 5–40
SYSTEM_CHECK system parameter, 5–12

T

Tape robots
automatic multipath failover, 4–13
TCP/IP Services for OpenVMS, 1–3
TECO editor
not available on I64 systems, 3–4
Terminal Fallback Facility (TFF), 4–20
restrictions, 4–21
TFF
See Terminal Fallback Facility
THREADCP command
behavior for I64, 5–32
Time zone changes, 4–21
Timer Queue Entries (TQEs), 5–41
TQEs
See Timer Queue Entries
Traceback facility
API Error (I64 only), 5–41
problem corrected, 5–41

U

UCB structure
updates to, 5–3
Upgrade
error when upgrading from V7.3-1 (Alpha only), 1–10
paths, 1–9

V

v*scanf and *snprintf namespace problem, 5–8
VAX Cluster Cache
See Virtual I/O Cache
VCC
See Virtual I/O Cache

VFC format sequential files, 5–36
VIOC
  See Virtual I/O Cache
Virtual I/O Cache (VIOC)
  not available on I64, 4–22
  superseded by XFC, 4–22
Visual Threads
  not supported for V8.2, A–3
VMS_SHOW
  DUMP_DEV errors (I64 Only), 4–4
Volume Shadowing for OpenVMS, 4–22 to 4–27
  ANALYZE/DISK/SHADOW command, 4–25
    and DDS, 4–26
  compatibility kits, 4–9
  device name requirement, 4–22
  dissimilar device shadowing (DDS)
    caution, 4–23
    KZPDC restriction, 4–24
  performance of merge operations, 4–25
  problem with dismount using /MINICOPY, 4–26
  Smart Array 5300 (KZPDC) restriction, 4–24
vsnprintf overwrite buffer, 5–7

W

Watchpoint Utility, 5–42
WEBES
  support on I64, 2–10
Whole program floating-point mode (I64 only), 5–42
Write bitmaps
  and dissimilar device shadowing (DDS), 4–23

X

X.25 data links unsupported (I64 only), 2–6
XA, 4–2
XFC
  See Extended File Cache

Z

zic updates, 5–10
ZLX graphics boards support, 6–21