HP OpenVMS Version 8.3 Release Notes

Order Number: BA322–90047

July 2006

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 8.3 OpenVMS Alpha Version 8.3

Hewlett-Packard Company
Palo Alto, California

© Copyright 2006 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, and 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.

ZK6677

The HP OpenVMS documentation set is available on CD.

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

Contents

Preface xv

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 Release Notes for OpenVMS for Integrity Servers Users 1–4
1.7.1 A6825A Gigabit Ethernet Controller and RX4640 Restriction V8.3 . . . 1–4
1.7.2 UI60 SCSI Support on the rx7620 and rx8620 1–5
1.7.3 Clearing the System Event Log (SEL) on Integrity Servers 1–5
1.7.4 Firmware for Integrity Servers 1–6
1.7.5 Booting from the Installation DVD 1–7
1.7.6 HP DECwindows Motif Release Notes 1–7
1.7.6.1 Keyboard Support 1–7
1.7.6.2 Connect Peripheral Devices Prior to Server Startup 1–8
1.7.6.3 Countdown Messages Displayed During Startup 1–8
1.8 Release Notes for OpenVMS Alpha Users 1–8
1.8.1 Firmware for OpenVMS Alpha Version 8.3 1–8
1.8.2 Upgrade Paths 1–9
1.9 CDSA: Secure Delivery Signing for Third Parties Available After Release of OpenVMS V8.3 1–10
1.10 Remove Any Lines from SYS$MANAGER:SYSTARTUP_VMS.COM That Start Encrypt or SSL 1–10
1.11 Kerberos for OpenVMS 1–10
1.12 Encryption for OpenVMS Installed with the OpenVMS Operating System 1–12
1.13 HP DECram V3.*n*: Remove Before Upgrading 1–13
1.14 Converting the LANCP Device Database After Upgrading 1–13
1.15 DECnet-Plus Requires a New Version 1–13
1.16 SHADOW_MAX_UNIT Default Setting and Memory Usage 1–14
1.17 TIE Kit Must be Removed Before Upgrade 1–15
1.18 Installation of a Layered Product to an Alternate Device or Directory May Fail 1–15

2 OpenVMS Associated Products Release Notes

2.1 Associated Product Support 2–1
2.2 CMAP Files Added 2–2
2.3 COBOL: Changes in I/O Run-Time Diagnostics and RMS Special Registers 2–2
2.4 COM for HP OpenVMS (Alpha Only) 2–2
2.4.1 COM for OpenVMS Support 2–2
2.4.2 Registry Access Error with Heavy Load of Applications 2–2
2.5 DECdfs Version 2.4 Required for OpenVMS Version 8.3 2–2
2.6 DEC PL/I: RTL Support for OpenVMS 2–3
2.7 FMS Kits 2–3
2.8 Graphical Configuration Manager (GCM) (Alpha Only) 2–4
2.9 HP DCE RPC for OpenVMS 2–4
2.9.1 DCE RPC Supports FAILSafe IP 2–4
2.9.2 Support for Tuning the Buffer Size of RPC Sockets 2–4
2.9.3 RTI (Remote Task Invocation) RPC (I64 Only) 2–4
2.9.4 Microsoft Lan Manager RPC Not Tested (I64 Only) 2–4
2.9.5 The utc_mulftime Factor Argument Type 2–4
2.9.6 Support for G_FLOAT and IEEE Floating-Point Types 2–5
2.10 HP DECnet-Plus Support for Mixed-Case Password 2–6
2.11 HP DECnet-Plus for OpenVMS: X.25 Data Links Not Supported (I64 Only) 2–6
2.12 HP DECram 2–6
2.12.1 DECram Ships With OpenVMS Version 8.2 2–7
2.12.2 Conflict with DECRYPT DCL Command 2–7
2.13 HP DECwindows Motif for OpenVMS 2–7
2.13.1 DECwindows Pause Screen Can Fail to Unlock (Alpha Only) 2–7
2.13.2 User-Written Transports Not Supported 2–7
2.14 HP Secure Web Server Version Support 2–8
2.15 HP Pascal for OpenVMS Alpha Systems 2–8
2.15.1 V5.8A (or Later) Required to Create STARLET Library (Alpha Only) 2–8
2.15.2 Installing HP Pascal After an Upgrade (Alpha Only) 2–9
2.16 WEBES and SEA Support on I64 Systems 2–9
2.17 NetBeans Version 3.6 Requires Java Standard Edition, Development Kit v 1.4.2-x 2–9

3 General User Release Notes

3.1 OpenVMS Freeware CDs 3–1
3.1.1 Freeware Menu Unavailable (I64 Only) 3–1
3.2 DCL Commands 3–2
3.2.1 CREATE/MAILBOX: Temporary Restriction 3–2
3.2.2 DIAGNOSE Command No Longer Supported 3–2
3.3 DECmigrate Not on Open Source Tools CD 3–2
3.4 HP Secure Web Browser 3–2
3.4.1 Increased Memory Required 3–2
3.4.2 Installation Error on ODS-2 Disk Volume (I64 Only) 3–2
3.5 Documentation Corrections 3–3
3.5.1 *HP OpenVMS System Analysis Tools Manual* 3–3
3.5.2 *HP OpenVMS Programming Concepts Manual* 3–3
3.5.2.1 Saving System Dumps 3–3
3.5.3 *HP OpenVMS Debugger Manual*: Client/Server Interface Correction 3–3

3.6 *HP OpenVMS DELTA/XDELTA Debugger Manual* Update 3–4
3.7 *HP OpenVMS Utility Routines Manual* Updated 3–4
3.7.1 *HP OpenVMS I/O User's Reference Manual*: PTD$READ Clarification 3–4
3.7.1.1 PTD$ Pseudoterminal Driver Correction 3–5
3.7.2 *HP OpenVMS Version 8.2 New Features and Documentation Overview*: Librarian Utility Corrections 3–5
3.7.2.1 /REMOVE Qualifier Correction 3–5
3.7.2.2 Accessing ELF Object Libraries Correction 3–5
3.7.3 *HP OpenVMS RTL Library (LIB$) Manual* Corrections 3–6
3.7.3.1 *HP OpenVMS RTL Library (LIB$) Manual*: Clarification of Rounding Rule for LIB$CVT_DX_DX 3–6
3.7.4 *HP OpenVMS RTL Library (LIB$) Manual*: Clarification of Platform Restrictions 3–7
3.7.5 *HP OpenVMS System Manager's Manual*: IPC Commands Restriction 3–7
3.7.5.1 $PUTMSG System Service Correction 3–8
3.7.6 *HP Volume Shadowing for OpenVMS*: Corrections to Memory Requirements 3–8
3.8 Network Update Restrictions from Version 8.2 to Version 8.2–1 3–8
3.9 Synchronous Data Links Not Supported 3–9
3.10 Duplex-Mode Mismatch Errors Reported by the LAN Drivers 3–9
3.11 Failure of AUDIT_SERVER to Initiate During Boot 3–10

4 System Management Release Notes

4.1 Monitor Utility Changes 4–1
4.1.1 Version-to-Version Compatibility of MONITOR Data 4–1
4.1.2 Playing Back Data from a Recording File 4–1
4.1.3 Monitoring Live Remote Data across a VMScluster 4–2
4.2 Updated Recommended File Security Attributes 4–2
4.3 System Management Notes 4–3
4.3.1 CIMSERVER Process Recommendation (I64 Only) 4–3
4.4 Recovering from System Hangs or Crashes (I64 Only) 4–3
4.5 DECdtm/XA with Oracle 8*i* and 9*i* (Alpha Only) 4–3
4.6 Device Unit Number Maximum Increased 4–4
4.7 ECP Data Collector and Performance Analyzer V5.5 (Alpha Only) 4–4
4.8 EDIT/FDL: Fixing Recommended Bucket Size 4–4
4.9 EFI$CP Utility: Use Not Recommended 4–4
4.10 EFI Shell Precautions on Shared or Shadowed System Disks 4–5
4.11 Error Log Viewer (ELV) Utility: TRANSLATE/PAGE Command 4–6
4.12 External Authentication 4–7
4.12.1 I64 External Authentication Support 4–7
4.12.2 SET PASSWORD Behavior Within a DECterm Terminal Session 4–7
4.12.3 No Password Expiration Notification on Workstations 4–7
4.13 OpenVMS Cluster Systems 4–8
4.13.1 OpenVMS I64 Cluster Support 4–8
4.13.2 Temporary Exceptions 4–8
4.13.3 Satellite Booting and LAN Failover 4–8
4.13.4 Creation of Error Log Dump File for Integrity Server Satellite Systems 4–9
4.14 Cluster Compatibility Patch Kits 4–9
4.14.1 Patch Kits Needed for Cluster Compatibility 4–10

4.14.2 API Can Correct Incompatibility of Fibre Channel and SCSI Multipath with Some Third-Party Products 4–11
4.14.3 DDT Intercept Establisher Routines and Device Configuration Notification Results 4–12
4.14.4 Cluster Performance Reduced with CI-LAN Circuit Switching 4–12
4.14.5 Multipath Tape Failover Restriction 4–13
4.14.6 No Automatic Failover for SCSI Multipath Medium Changers 4–13
4.14.7 Availability Manager AVAIL_MAN_BASE Kit Fails to Collect Lock Contention Information 4–14
4.15 OpenVMS Galaxy (Alpha Only) 4–14
4.15.1 Galaxy Definitions 4–14
4.16 Multiple nPartitions on Cell-based Systems 4–14
4.16.1 OpenVMS Graphical Configuration Manager 4–15
4.16.2 Galaxy on ES40: Uncompressed Dump Limitation 4–15
4.16.3 Galaxy on ES40: Turning Off Fast Path 4–15
4.17 OpenVMS Management Station 4–15
4.18 OpenVMS Registry Can Corrupt Version 2 Format Database 4–15
4.19 System Parameters 4–16
4.19.1 New System Parameters 4–16
4.19.2 Obsolete System Parameters 4–16
4.19.3 System Parameter Changes 4–17
4.19.4 Documentation Error: LCKMGR_CPUID System Parameter 4–17
4.19.5 MMG_CTLFLAGS: Documentation Error 4–17
4.20 Terminal Fallback Facility (TFF) 4–17
4.21 User Environment Test Package (UETP) (I64 Only) 4–19
4.22 Recommended Caching Methods 4–19
4.23 Volume Shadowing for OpenVMS 4–19
4.23.1 Device Name Requirement 4–19
4.23.2 Warning About Using SET SHADOW and SHOW SHADOW in DCL Command Procedures 4–20
4.23.3 Write Bitmaps and Dissimilar Device Shadowing (DDS) Caution 4–20
4.23.4 KZPDC (Smart Array 5300) Restrictions 4–21
4.23.5 Changes in Shadow Set Merge Delay Computation 4–22
4.23.6 Dismount of Shadow Set Member Using /MINICOPY 4–22
4.24 Authentication and Credential Management Extension (ACME) 4–23
4.24.1 New and Changed Features 4–23
4.25 Default Startup Order Required for OpenVMS and Kerberos ACME Agents 4–24
4.26 Traceback API Problem Fixed 4–24
4.27 WBEM Services for OpenVMS Version 2.0 Release Notes 4–24
4.27.1 Based on OpenPegasus 2.5 4–25
4.27.2 Supports nPartitions and iCAP 4–25
4.27.3 Not Designed for Homogeneous Clusters 4–25
4.27.4 Restart cimserver.exe to Unload Providers on OpenVMS 4–25
4.27.5 Use Quotes Around Command Line Options 4–25
4.27.6 Clean Up WBEMCIM Repository Directory Tree After Uninstall 4–25
4.28 Missing $SIGNAL_ARRAY_64 System Service on I64 4–25
4.29 *HP OpenVMS System Analysis Tools Manual* 4–26
4.29.1 Definitions of Bits in DUMPSTYLE 4–26
4.29.2 Saving System Dumps 4–26
4.29.3 Invoking SDA When Rebooting the System 4–27
4.29.4 SDA CONTEXT 4–27
4.29.5 SDA Symbol Initiation 4–27
4.29.6 ANALYZE 4–27

4.29.7 Crash_Dump 4–28
4.29.8 SDA Commands 4–28
4.29.9 Copy Command 4–28
4.29.10 DUMP Command 4–28
4.29.11 EVALUATE Command 4–29
4.29.12 MAP Command 4–29
4.29.13 SET CPU Command 4–29
4.29.14 SET PROCESS Command 4–30
4.29.15 READ Command 4–30
4.29.16 SEARCH Command 4–30
4.29.17 SHOW_CALL_FRAME Command 4–30
4.29.18 SHOW CLUSTER Command 4–31
4.29.19 SHOW CRASH Command 4–31
4.29.20 SHOW DUMP Command 4–31
4.29.21 SHOW PROCESS Command 4–32
4.29.22 SHOW RESOURCES Command 4–33
4.29.23 SHOW SPINLOCKS Command 4–33
4.29.24 SHOW MEMORY Command 4–34
4.29.25 SHOW GCT Command 4–34
4.29.26 SHOW CPU Command 4–34
4.29.27 SHOW SYMBOL Command 4–34
4.29.28 SHOW UNWIND Command 4–34
4.29.29 SHOW SWIS Command 4–35
4.29.30 CLUE CONFIG Command 4–35
4.29.31 CLUE REGISTER Command 4–35
4.29.32 I64 ISD_Labels Index Table 4–36
4.29.33 Compiling and Linking an SDA Extension 4–36
4.29.34 Debugging an Extension 4–36
4.29.35 Callable Routines Overview 4–36
4.29.36 Setting Up the Target System for Connections 4–37
4.29.37 OpenVMS Alpha System Dump Debugger 4–47
4.29.38 OpenVMS Alpha Watchpoint Utility 4–49
4.29.39 SET PROCESS/LOG Command 4–50

5 Programming Release Notes

5.1 System Service Changes 5–1
5.1.1 Additions 5–1
5.2 Range of Job-Limit Item Codes Increased for $GETQUI and $SNDJBC System Services 5–2
5.3 Privileged Programs Might Need to Be Recompiled (Alpha Only) 5–2
5.4 Privileged Data Structures Updates 5–2
5.4.1 KPB Extensions 5–3
5.4.2 CPU Name Space 5–3
5.4.3 64-Bit Logical Block Number (LBN) 5–3
5.4.4 Forking to a Dynamic Spinlock 5–3
5.4.5 UCB/DDB Updates 5–4
5.4.6 PCB$T_TERMINAL Size Increase 5–4
5.4.7 Per-Thread Security Impacts Privileged Code and Device Drivers 5–5
5.5 Applications Using Floating-Point Data 5–6
5.5.1 IEEE Floating-Point Filter (I64 Only) 5–6
5.5.2 Limitation When Using Ctrl/C and STOP Button (OpenVMS Alpha) 5–7
5.5.3 Ada Event Support (I64 Only) 5–7

5.5.4 SHOW SYMBOL/TYPE Now Reports Correct Array Size (Alpha and I64) . . . 5–7
5.5.5 EXAMINE/INSTRUCTION %PREVLOC Command Is Fixed (I64 Only) . . . 5–7
5.5.6 SHOW MODULE Command Now Computes Module Size (I64 only) . . . 5–8
5.5.7 C++ Language Issues (I64 Only) . . . 5–8
5.6 Ada Compiler(I64 Only) . . . 5–8
5.7 Backup API: Journaling Callback Events Restriction . . . 5–8
5.8 C Programs: Compiling with CASE_LOOKUP=SENSITIVE Settings . . . 5–8
5.9 C Run-Time Library . . . 5–9
5.9.1 C RTL TCP/IP Header File Updates . . . 5–9
5.9.2 toascii Function Added . . . 5–9
5.9.3 64-Bit sigaction Problem Fixed . . . 5–10
5.9.4 64-Bit Pointer Capability Added to Several Math Functions . . . 5–10
5.9.5 2-GB malloc No Longer Fails Silently . . . 5–10
5.9.6 Memory Leak in exec* Fixed . . . 5–10
5.9.7 Behavior of exit after a Failed execl Fixed . . . 5–10
5.9.8 confstr Enhancements . . . 5–11
5.9.9 fopen Failure Fixed . . . 5–11
5.9.10 Possible File-Pointer-Locking Hang Condition . . . 5–11
5.9.11 Backport Library No Longer Shipped . . . 5–11
5.9.12 Header File <time.h> Changes . . . 5–11
5.9.13 Header File <time.h> Makes *_r Non-ANSI Functions Visible . . . 5–12
5.9.14 Header File <decc$types.h>: time_t int Declaration . . . 5–12
5.9.15 New DECC$SHRP.EXE Image . . . 5–12
5.9.16 Header File <wchar.h> and C++ %CXX-W-ENVIRSTKDIRTY Message . . . 5–13
5.9.17 Header File <builtins.h> __CMP_SWAP* and _Interlocked* Visible to C++ . . . 5–13
5.9.18 Extra Parameters to fcntl Ignored . . . 5–13
5.9.19 Problem with fwrite to stdout Fails with Large System MAXBUF . . . 5–13
5.9.20 Problem with Read/Write for Socket Transfers Greater Than 64K . . . 5–13
5.9.21 Problem with Nanosleep on I64 Systems . . . 5–13
5.9.22 Builtin __fci Added for I64 Systems . . . 5–14
5.9.23 _FAST_TOUPPER Macro Added . . . 5–14
5.9.24 Call to atof("NaN") no Longer Gives Arithmetic Trap . . . 5–14
5.9.25 No New Entries for DECC$*.OLB Object Libraries . . . 5–14
5.10 Calling Standard and Rotating Registers (I64 Only) . . . 5–14
5.11 Common Data Security Architecture (CDSA) Considerations . . . 5–15
5.11.1 Secure Delivery . . . 5–15
5.11.2 Installation and Initialization Considerations . . . 5–15
5.12 Debugging Modes: Avoiding CPUSPINWAIT Bugchecks . . . 5–16
5.13 Delta/XDelta Debuggers . . . 5–16
5.13.1 XDELTA Register Display Consideration (I64 Only) . . . 5–16
5.14 File Applications: Corrections to *Guide to OpenVMS File Applications* . . . 5–16
5.15 HP BLISS Compiler Warnings with RMS Structures (I64 Only) . . . 5–17
5.16 Potential Must-Be-Zero RMS Error: Making Room for New File Options in the FAB . . . 5–18
5.17 HP COBOL Run-Time Library (RTL) . . . 5–18
5.18 HP Fortran for I64 . . . 5–18
5.19 HP MACRO for OpenVMS . . . 5–19
5.19.1 HP MACRO for OpenVMS I64 . . . 5–19

5.19.2 HP MACRO for OpenVMS Alpha Systems 5–20
5.19.3 /TIE Qualifier Defaults Differ on Alpha and I64 5–21
5.19.4 /OPTIMIZE=VAXREGS Qualifier Not Supported on I64 5–21
5.19.5 Floating Divide-by-Zero Error Not Raised (I64 Only) 5–21
5.20 Hypersort Utility 5–21
5.20.1 Reporting a Problem to HP 5–21
5.20.2 Large Files Restriction 5–22
5.20.3 Hypersort and VFC Files Restriction 5–22
5.20.4 /FORMAT=RECORD_SIZE Restriction 5–22
5.20.5 Using Hypersort with Search Lists and Other Uses of Logical Names 5–22
5.20.6 Lack of Free Space for Work Files 5–22
5.20.7 Input Asterisk (*) Restriction 5–22
5.20.8 Optimal Working Set Extent and Page File Quota Settings 5–23
5.21 Intel® Assembler (I64 Only) 5–23
5.22 Librarian Utility 5–23
5.22.1 Linking Against Data-Reduced ELF Object Libraries Not Recommended (I64 Only) 5–23
5.22.2 Failure to Insert or Replace .STB files in an I64 Library (I64 Only) 5–23
5.22.3 Librarian Fails to Report Errors When Process Quota Too Low 5–24
5.23 Analyze Utility for OpenVMS (I64 Only) 5–24
5.23.1 Releasing Process Resources Fixed in I64 Analyze Image /SELECT 5–24
5.23.2 Selective Output for Debug Line Information Fixed 5–24
5.24 Command Definition Utility (I64 Only) 5–24
5.24.1 Record Attributes Fixed for I64 Images 5–25
5.25 Linker Utility for OpenVMS Alpha 5–25
5.25.1 Linker Appears to Hang When Many Files Are Specified 5–25
5.25.2 Change in Linker Default Behavior with Library Check 5–25
5.25.3 Limit of 25 Elements on Stack 5–26
5.26 Linker Utility for OpenVMS I64 5–26
5.26.1 Linker Writes Incorrect Interimage Debug Fixups into Debug Symbol File 5–26
5.26.2 OpenVMS I64 Object Module and Image File Information Currently Unavailable 5–27
5.26.3 /SELECTIVE_SEARCH Might Incorrectly Ignore Transfer Address 5–27
5.26.4 Differences Between the I64 Linker and the Alpha Linker 5–28
5.26.5 LINK_ORDER Section Header Flag Not Supported 5–28
5.26.6 Linking Against Data-Reduced ELF Object Libraries Not Recommended 5–28
5.26.7 Error in Handling Initialized Overlaid Program Sections Fixed 5–29
5.26.8 Removal of Linker Qualifiers /EXPORT_SYMBOL_VECTOR and /PUBLISH_GLOBAL_SYMBOLS 5–29
5.26.9 Support for Longer Symbol Names in Options 5–29
5.26.10 Better Use of Memory for Linker-Created Code Stubs 5–29
5.26.11 Compiler Support for Demangled Symbol Names 5–29
5.26.12 Maximum Number of Sections 5–30
5.26.13 Incorrect Creation Date of Shareable Images in the Map File 5–30
5.27 LTDRIVER: CANCEL SELECTIVE Restriction 5–30
5.28 Mail Utility: Threads Restriction for Callable Mail 5–30
5.29 OpenVMS Debugger 5–30

5.29.1 STEP/SEMANTIC_EVENT Information Missing in Debugger Command-Line Help . . . 5–31
5.29.2 Problems and Conditions Corrected in this Release (I64 Only) . . . 5–31
5.29.3 General Conditions and Workarounds (I64 Only) . . . 5–31
5.29.4 BASIC Language Issues (I64 Only) . . . 5–32
5.29.5 C++ Language Issues (I64 Only) . . . 5–32
5.29.6 COBOL Language Issues (I64 Only) . . . 5–32
5.29.7 Fortran Language Issues (I64 Only) . . . 5–32
5.29.8 Pascal Language Issues (I64 Only) . . . 5–32
5.29.9 SET SCOPE Command: Behavior Change . . . 5–33
5.29.10 SHOW IMAGE Command Change . . . 5–33
5.29.11 Client/Server Interface: Previous Versions Not Supported (Alpha Only) . . . 5–33
5.30 OpenVMS System Dump Analyzer (SDA) . . . 5–33
5.30.1 CLUE Commands Not Ported to OpenVMS I64 . . . 5–33
5.31 PL/I Libraries Not Included in OpenVMS I64 Version 8.2 . . . 5–34
5.32 POSIX Threads Library . . . 5–34
5.32.1 Stack Overflows During Exception Handling (I64 Only) . . . 5–34
5.32.2 THREADCP Command Behavior on I64 Systems . . . 5–35
5.32.3 Floating-Point Compilations and Exceptions (I64 Only) . . . 5–35
5.32.4 C Language Compilation Header File Changes . . . 5–35
5.32.5 New Priority Adjustment Algorithm . . . 5–36
5.32.6 Process Dumps . . . 5–36
5.32.7 Dynamic CPU Configuration Changes . . . 5–36
5.32.8 Debugger Metering Function Does Not Work . . . 5–37
5.33 RTL Library (LIB$) . . . 5–37
5.33.1 RTL Library (LIB$) Help Omission . . . 5–37
5.33.2 RTL Library (LIB$): Calling Standard Routines (I64 Only) . . . 5–37
5.34 Screen Management (SMG$) Documentation . . . 5–38
5.35 SORT32 Utility . . . 5–39
5.35.1 CONVERT Problem With DFS-Served Disks . . . 5–39
5.35.2 Temporary Work Files Not Always Deleted . . . 5–39
5.35.3 SORT/SPECIFICATION With Compound Conditions: Requirement . . . 5–39
5.35.4 Performance Problem with Variable Length Records . . . 5–39
5.35.5 Work File Directories Restriction . . . 5–40
5.36 System Services . . . 5–40
5.37 Timer Queue Entries (TQEs) . . . 5–40
5.38 Watchpoint Utility (I64 Only) . . . 5–40
5.39 Whole Program Floating-Point Mode (I64 Only) . . . 5–40
5.40 HP OpenVMS Debugger Heap Analyzer Conditions and Workarounds (I64 only) . . . 5–41

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 RAD Support . . . 6–4
6.4.3 License Requirements . . . 6–4
6.4.4 STOP/CPU and Shutdown Behavior . . . 6–4
6.4.5 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 DECW$OPENGLSHR_RADEON.EXE Renamed to DECW$MESA3DSHR_RADEON.EXE . . . 6–9
6.9.3 Support Limitations . . . 6–9
6.9.4 Video Artifacts at High Refresh Rates . . . 6–10
6.9.5 OpenGL Supports IEEE Arithmetic Only . . . 6–10
6.9.6 DECwindows Server Hangs When Output Is Written to the Graphics Console (OPA0) . . . 6–10
6.9.7 Monitor Must Be Connected During Initialization . . . 6–10
6.9.8 Boot Reset Recommendation (Alpha Only) . . . 6–10
6.9.9 No Overlay Planes . . . 6–11
6.9.10 Single Colormap . . . 6–11
6.9.11 Single Bit Depth for All Windows . . . 6–11
6.9.12 Pixel Depth for Read/Write Color Map . . . 6–11
6.9.13 Backing Store/Save Unders Not Supported for 3D Windows . . . 6–12
6.9.14 Threads Restriction . . . 6–12
6.9.15 No Support for Single Buffered Visuals . . . 6–12
6.9.16 No 3D Support for Color Index Mode . . . 6–12
6.9.17 Timer Mechanism . . . 6–12
6.10 DECwindows X11 Display Server (Alpha Only) . . . 6–13
6.10.1 S3 Multihead Graphics . . . 6–13
6.10.2 Multiple USB Keyboards and Mice Not Supported on DECwindows on OpenVMS V8.3 . . . 6–13
6.11 DIGITAL Modular Computing Components (DMCC) . . . 6–13
6.11.1 Alpha 5/366 and 5/433 PICMG SBC Restriction . . . 6–13
6.11.2 Updating the SRM Console . . . 6–13
6.12 Digital Personal Workstation: Booting OpenVMS V7.3-1 and Higher . . . 6–14
6.13 Dual-Controller HSG*nn* with Many LUNs Can Fail Under Heavy I/O Load . . . 6–14
6.14 Open3D Graphics Licensing Change . . . 6–14
6.15 PowerStorm 300/350 PCI Graphics Support for OpenVMS . . . 6–15
6.16 RF*nn* DSSI Disk Devices and Controller Memory Errors . . . 6–16
6.17 RZ*nn* Disk Drive Considerations . . . 6–18
6.17.1 RZ25M and RZ26N Disk Drives: Recommendations . . . 6–18
6.17.2 RZ26N and RZ28M Disks: Recommended Firmware Support . . . 6–18

6.17.3 RZ26L and RZ28 Disks: Required Firmware for Multihost Use 6–18
6.17.3.1 Firmware Revision Level 442 Requirements 6–19
6.17.3.2 Firmware Revision Level 442 Installation Procedure 6–19
6.18 sx1000 Integrity Superdome 6–19
6.19 ZLX Graphics Boards Support 6–20
6.20 Recompiling and Relinking OpenVMS Device Drivers 6–20
6.20.1 Alpha and VAX SCSI Device Drivers 6–20
6.20.2 OpenVMS Alpha Device Drivers 6–20
6.21 Device Driver MON Version Handling 6–21
6.22 Possible Per-Threads Security Impact on Alpha Device Drivers 6–21
6.23 Device IPL Setup for OpenVMS Alpha Drivers 6–21
6.24 CRCTX Routines Enhanced 6–21
6.25 Adapter Release Notes 6–22
6.25.1 Fibre Channel EFI Driver and Firmware Requirements 6–22
6.25.2 Booting with Multiple Fibre Channel Boot Entries 6–22

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 NetBeans Version 3.6 Support Ends with OpenVMS Version 8.3 . . . . . 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
1–2 Firmware Versions for Entry-Class Integrity Servers 1–6
4–2 Patch Kits Required for Cluster Compatibility 4–10
4–2 Patch Kits Required for Cluster Compatibility 4–11
4–3 Galaxy Definitions 4–14
4–4 TFF Character Fallback Tables 4–18
4–5 I64 ISD_Labels Index 4–36
6–1 Changes to Device Description Block 6–7

6–2 Supported Microcode Revision Levels 6–16
6–3 Commands for Updating Microcode in Certain DSSI Disk Devices . . . 6–17
6–4 Revision Level 442 Firmware Compatibility 6–19
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 for I64 Integrity servers Version 8.3 operating system. Read this manual before you install, upgrade, or use OpenVMS Version 8.3.

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, *V8.2*) or the version when an old note was last updated (for example, a note from V8.2 that was revised for Version 8.3 will be labeled with *V8.3*).

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 8.3 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).
I64	An abbreviation representing HP OpenVM for Integrity servers
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 or upgrading to OpenVMS Version 8.3. 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 8.3:

- *HP OpenVMS Version 8.3 Release Notes* (this manual)
- *HP OpenVMS Version 8.3 New Features and Documentation Overview*
- *HP OpenVMS Version 8.3 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
- HP DECram 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

V8.3

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 8.3, you have the option of installing the following supported HP networking software:

- Either HP DECnet-Plus Version 8.3 for OpenVMS or HP DECnet Phase IV Version 8.3 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.

Alternatively, after you install OpenVMS, you can install your choice of another third-party networking product that runs on OpenVMS Version 8.3.

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; printed manuals can be ordered from HP (800-282-6672).

HP strongly recommends that you apply the current Engineering Change Order (ECO) for TCP/IP Services for OpenVMS Version 5.5 after upgrading to OpenVMS Version 8.2–1 to ensure that the latest software is installed.

1.5 Disk Incompatibility with Older Versions of OpenVMS

V8.3

The OpenVMS Version 8.3 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 8.3 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

V8.3

The following table lists which versions of DECwindows Motif are supported on various platforms of the OpenVMS Version 8.3 operating system.

Table 1–1 Supported Versions of DECwindows Motif

OpenVMS Version	DECwindows Motif Version
OpenVMS I64 8.3	DECwindows Motif for OpenVMS I64 V1.5
OpenVMS Alpha 8.3	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 Release Notes for OpenVMS for Integrity Servers Users

The following notes are primarily of interest to users of OpenVMS for Integrity servers.

1.7.1 A6825A Gigabit Ethernet Controller and RX4640 Restriction V8.3

V8.3

The HP sx1000 Chipset for HP Integrity servers provides the CPU, memory, and I/O subsystem to the HP Integrity rx7620, HP Integrity 8620, and HP Integrity Superdome servers. The cell controller is combined with four CPU chips into the computing cell in the sx1000 Chipset architecture. The cell controller chip also provides paths to the I/O devices and off-cell memory.

The rx7620, rx8620, and Superdome servers provide a varying number of sx1000 Chipset cells. The rx7620 provides up to 2 cells (8 CPUs), the rx8620 provides up to 4 cells (16 CPUs), and the Superdome provides up to 16 cells (64 CPUs).

For OpenVMS I64 Version 8.3, 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 8.3, and reached end-of-life in 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 8.3, and customers using them must upgrade to the A6826A FC adapter before running production applications on Version 8.2.

1.7.2 UI60 SCSI Support on the rx7620 and rx8620

V8.3

The rx7620 and rx8620 have an internal U160 (SCSI) that comes with the system as core I/O. The internal connections to the racks of SCSI disks (which appear on the front of the system box) are supported by OpenVMS. The internal box also has two external ports. HP does not support attaching them (via cables) to a SCSI rack.

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

V8.3

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 8.3 Upgrade and Installation Manual*.

1.7.4 Firmware for Integrity Servers

V8.3

OpenVMS I64 Version 8.3 was tested with the latest firmware for each of the supported Integrity servers. The following table lists the recommended firmware versions:

Table 1–2 Firmware Versions for Entry-Class Integrity Servers

System	System Firmware	BMC Firmware	MP Firmware	DHCP Firmware
rx1600	2.18	3.48	E.03.15	N/A
rx1620	2.18	3.48	E.03.15	N/A
rx2600	2.31	1.53	E.03.15	N/A
rx2620	3.17	3.47	E.03.15	N/A
rx4640	3.17	3.49	E.03.15	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 8.3 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 8.3 Upgrade and Installation Manual*.

1.7.5 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.

1.7.6 HP DECwindows Motif Release Notes

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

1.7.6.1 Keyboard Support

V8.2

The only model keyboard supported on HP DECwindows Motif for OpenVMS I64 systems is the LK463 (AB552A for I64) keyboard. Although other types of keyboards may function in the OpenVMS I64 environment, HP does not currently support them.

1.7.6.2 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.

1.7.6.3 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.

1.8 Release Notes for OpenVMS Alpha Users

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

1.8.1 Firmware for OpenVMS Alpha Version 8.3

V8.3

OpenVMS Alpha Version 8.3 was tested with the platform-specific firmware included on Alpha Systems Firmware CD Version 6.8. For older platforms no longer included on the Firmware CD, 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 8.3 kit includes the Alpha Systems Firmware CD-ROM and Release Notes. Read the firwmware 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):

`http://ftp.digital.com/pub/DEC/Alpha/firmware/`

1.8.2 Upgrade Paths

V8.3

You can upgrade *directly* to OpenVMS Alpha Version 8.3 from only the following versions of OpenVMS Alpha.

For Alpha:

Version 7.3-2 to V8.3
Version 8.2 to V8.3

For I64:

Version 8.2 to V8.3
Version 8.2-1 to V8.3

If you are currently running OpenVMS Alpha Version 6.2*x* through 7.3, inclusive, you must first upgrade to Version 7.3-2, and then to Version 8.3. Note that standard support for OpenVMS Alpha Version 7.3-1 systems ended 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 8.3:

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 4.14.1.

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 8.3. For

more information, refer to the *HP OpenVMS Version 8.3 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 8.3, and for instructions on installing OpenVMS I64 Version 8.3, refer to the *HP OpenVMS Version 8.3 Upgrade and Installation Manual*.

1.9 CDSA: Secure Delivery Signing for Third Parties Available After Release of OpenVMS V8.3

V8.3

CDSA Version 2.2, which is included with OpenVMS Version 8.3, will allow for organizations other than OpenVMS Engineering to sign and validate OpenVMS kits using the new OpenVMS Secure Delivery functionality. Although the capability to sign and validate kits is included with CDSA Version 2.2, the supporting business practices are not currently in place to support third party vendors signing their own kits. HP expects to allow third party signing soon. Please monitor the HP Developer & Solution Partner Program web site (http://www.hp.com/dspp) for announcements about the availability of this program.

1.10 Remove Any Lines from SYS$MANAGER:SYSTARTUP_VMS.COM That Start Encrypt or SSL

V8.3

The startup command procedures for Encrypt and SSL are now called from the VMS$LPBEGIN-050_STARTUP.COM procedure. If you are upgrading from a previous version of OpenVMS that had these products installed, edit SYSTARTUP_VMS.COM to remove the calls to SYS$STARTUP:ENCRYPT_START.COM and SYS$STARTUP:SSL$STARTUP.COM. This will prevent these command procedures from executing twice.

1.11 Kerberos for OpenVMS

V8.3

Before configuring or starting Kerberos, check the HP TCP/IP local host database to determine whether your hostname definition is the short name (for example, node1) or the fully qualified domain name (FQDN) (for example, node1.hp.com).

If your host name definition is the short name, you must run TCPIP$CONFIG to change the definition to the fully qualified name.

The following example shows that the hostname is the short name:

```
$ TCPIP SHOW HOST/LOCAL NODE1

     LOCAL database

Host address    Host name

1.2.3.4  node1
```

The following log is an example of how to change the host name definition to the FQDN.

```
$ @SYS$STARTUP:TCPIP$CONFIG

                TCP/IP Network Configuration Procedure

        This procedure helps you define the parameters required
        to run HP TCP/IP Services for OpenVMS on this system.

        Checking TCP/IP Services for OpenVMS configuration database files.

        HP TCP/IP Services for OpenVMS Configuration Menu

        Configuration options:

                 1  -  Core environment
                 2  -  Client components
                 3  -  Server components
                 4  -  Optional components

                 5  -  Shutdown HP TCP/IP Services for OpenVMS
                 6  -  Startup HP TCP/IP Services for OpenVMS
                 7  -  Run tests

                 A  -  Configure options 1 - 4
                [E] -  Exit configuration procedure

Enter configuration option: 1

        HP TCP/IP Services for OpenVMS Core Environment Configuration Menu

        Configuration options:

                 1  -  Domain
                 2  -  Interfaces
                 3  -  Routing
                 4  -  BIND Resolver
                 5  -  Time Zone

                 A  -  Configure options 1 - 5
                [E] -  Exit menu

Enter configuration option: 2

      HP TCP/IP Services for OpenVMS Interface & Address Configuration Menu

 Hostname Details: Configured=node1, Active=node1

 Configuration options:

   1  -  WE0 Menu (EWA0: TwistedPair 1000mbps)
   2  -  1.2.3.4/21    node1               Configured,Active

   3  -  IE0 Menu (EIA0: TwistedPair 100mbps)

   I  -  Information about your configuration

  [E] -  Exit menu

Enter configuration option: 2

      HP TCP/IP Services for OpenVMS Address Configuration Menu

      WE0 1.2.3.4/21 node1 Configured,Active WE0

 Configuration options:

         1  - Change address
         2  - Set "node1" as the default hostname
         3  - Delete from configuration database
         4  - Remove from live system
         5  - Add standby aliases to configuration database (for failSAFE IP)

        [E] - Exit menu
```

```
Enter configuration option: 1

    IPv4 Address may be entered with CIDR bits suffix.
    E.g. For a 16-bit netmask enter 10.0.1.1/16

Enter IPv4 Address [1.2.3.4/21]:
Enter hostname [node1]: node1.hp.com

Requested configuration:

      Address  : 1.2.3.4/21
      Netmask  : 255.255.248.0 (CIDR bits: 21)
      Hostname : node1.hp.com

* Is this correct [YES]:

  "node1" is currently associated with address "1.2.3.4".
  Continuing will associate "node1.hp.com" with "1.2.3.4".

* Continue [NO]: YES
Deleted host node1 from host database
Added hostname node1.hp.com (1.2.3.4) to host database
* Update the address in the configuration database [NO]: YES
Updated address WE0:1.2.3.4 in configuration database
* Update the active address [NO]: YES
WE0: delete active inet address node1.hp.com
Updated active address to be WE0:1.2.3.4
```

Next, type E three times to exit the TCP/IP Services configuration menus and return to the DCL ($) prompt.

To verify your change, enter the following command:

```
$ TCPIP SHOW HOST/LOCAL NODE1

     LOCAL database

Host address    Host name

1.2.3.4  node1.hp.com
```

If you have not previously configured an earlier version of Kerberos on your system, or if you changed your TCP/IP hostname definition to the FQDN as shown above, you must run the Kerberos configuration program before you start Kerberos.

To reconfigure Kerberos, enter the following command:

```
$ @SYS$STARTUP:KRB$CONFIGURE
```

After you have a valid configuration, start Kerberos with the following command:

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

For more information, see the *Kerberos for OpenVMS Installation Guide and Release Notes*.

1.12 Encryption for OpenVMS Installed with the OpenVMS Operating System

V8.3

When you install or upgrade OpenVMS, Encryption for OpenVMS creates its own ENCRYPT and DECRYPT commands. Encryption for OpenVMS starts automatically (after SSL for OpenVMS, which also starts automatically). For more information about Encryption for OpenVMS, see *HP OpenVMS Version 8.3 New Features and Documentation Overview*.

Note

With Version 8.3 of OpenVMS, the DCL command DECRAM is removed because it conflicts with the new DECRYPT command (DECRYPT overwrites the default definition of DECRAM, which you might have been using to start DECram). You should update any command procedures that use the DECRAM command so that they use the foreign command style of DCL. For example:

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

This change affects the use of the DCL command only; all other aspects of the DECram product remain the same. If you have older versions of DECram on your OpenVMS Alpha system, you must remove them before upgrading. See Section 1.13.

1.13 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). If you upgrade to Version 8.3 on an OpenVMS Alpha system from OpenVMS Version 7.3-2, you must remove any old versions of DECram. Refer to the *HP OpenVMS Version 8.3 Upgrade and Installation Manual* for details.

More DECram release notes are included in Section 2.12.

1.14 Converting the LANCP Device Database After Upgrading

V8.3

When you upgrade to OpenVMS Alpha Version 8.3 from OpenVMS Version 7.3-2, you might also need to convert the LAN device database to the Version 8.3 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
```

1.15 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 make this necessary is a change of behavior in AUTOGEN that was introduced in Version 7.3-2.

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. Because 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*.

1.16 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.

1.17 TIE Kit Must be Removed Before Upgrade

V8.2-1

The Translated Image Environment (TIE) has been integrated into OpenVMS I64 Version 8.2–1. Refer to the HP OpenVMS Systems Migration Software Web site for further information.

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

If you have installed any version of the Translated Image Environment (TIE) PCSI kit (HP-I64VMS-TIE) on OpenVMS I64 Version 8.2 or Version 8.2-1, you must manually remove the TIE kit before you upgrade to OpenVMS I64 Version 8.3.

Use the following command to remove the TIE product kit:

```
$ PRODUCT REMOVE TIE
```

Do not install the TIE product kit, HP I64VMS TIE V1.0, on OpenVMS I64 Version 8.2-1 or Version 8.3.

1.18 Installation of a Layered Product to an Alternate Device or Directory May Fail

V8.3

By default the `PRODUCT INSTALL` command installs a layered product on the system device in the `SYS$COMMON` directory tree. If you choose to install a layered product to an alternate device or directory using the `/DESTINATION=dev:[dir]` qualifier (or by defining the logical name `PCSI$DESTINATION`), the installation may fail with an error message stating that one of the following files cannot be found: `[SYSLIB]DCLTABLES.EXE`, `[SYSHLP]HELPLIB.HLB`, or `[SYSLIB]STARLET*.*`. If this happens, answer YES to the question, "Do you want to terminate? [YES]," and then retry the installation using the `/NORECOVERY_MODE` qualifier.

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.

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 V1.6 is available on the latest consolidated layered product CD.

2.2 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

2.3 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.

2.4 COM for HP OpenVMS (Alpha Only)

The following release notes pertain to COM for HP OpenVMS.

2.4.1 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/`

2.4.2 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.

2.5 DECdfs Version 2.4 Required for OpenVMS Version 8.3

V8.3

DECdfs Version 2.4 is required for OpenVMS Version 8.3. If you try to use an older version of DECdfs, you will get an error message.

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.

2.6 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 5.31.

2.7 FMS Kits

V8.3

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

FMS V2.4 is supported on OpenVMS Alpha V8.3.

FMS V2.5 is supported on OpenVMS V8.2 and later systems (Alpha and I64).

2.8 Graphical Configuration Manager (GCM) (Alpha Only)

The Graphical Configuration Manager (GCM) is included on the Layered Products CD 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/
```

2.9 HP DCE RPC for OpenVMS

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

2.9.1 DCE RPC Supports FAILSafe IP

V8.2

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

2.9.2 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.

2.9.3 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.

2.9.4 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.

2.9.5 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`.

2.9.6 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.

2.10 HP DECnet-Plus Support for Mixed-Case Password

V8.3

OpenVMS Versions 7.3-2 and higher allow use of mixed-case and extended characters in passwords when PWDMIX is specified as login flags. Prior to DECnet-Plus V8.3, DECnet did not support mixed-case passwords for task-to-task communication and remote file access using DECnet.

2.11 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.

2.12 HP DECram

This section contains release notes pertaining to DECram.

Note

Refer to Section 1.13 for more information on HP DECram.

2.12.1 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.

2.12.2 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 HP DECwindows Motif for OpenVMS

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

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

V8.3

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)

2.13.2 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, TCP/IP, 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.

2.14 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. 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.

2.15 HP Pascal for OpenVMS Alpha Systems

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

2.15.1 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.

2.15.2 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.

2.16 WEBES and SEA Support on I64 Systems

V8.3 The latest version of WEBES (WEBased Enterprise Services) can be obtained from the WEBES homepage at:

`http://h18000.www1.hp.com/support/svctools/`

2.17 NetBeans Version 3.6 Requires Java Standard Edition, Development Kit v 1.4.2-x

V8.3

NetBeans Version 3.6 for OpenVMS Alpha and OpenVMS I64 is supported only on Java Platform, Standard Edition, Development Kit (JDK) v 1.4.2-x.

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 8.3 New Features and Documentation Overview*.

3.1 OpenVMS Freeware CDs

V8.3

Included in the OpenVMS Version 8.3 media kit are the OpenVMS Freeware Version 8.0 CDs. The Freeware CDs contain free software tools and utilities for creating applications and for using and managing OpenVMS systems.

To mount the Freeware CDs, insert one of the CD volumes into the CD drive and enter the following command to mount and display the contents of the Freeware volume.

```
$ MOUNT/OVERRIDE=IDENTIFICATION ddcu:
```

In this command, the *ddcu:* specification represents the device name of the CD or DVD device on your OpenVMS system. This device name is specific to each OpenVMS system.

```
$ TYPE ddcu:[FREEWARE]FREEWARE_README.TXT
```

Duplicate copies of this file are found on each volume of the Freeware 8.0 distribution, and you can view its contents by using the `TYPE` command or your preferred text editor.

For additional information about the Freeware, refer to the `FREEWARE_README.TXT` files.

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

The [FREEWARE]FREEWARE.COM Freeware menu system interface has been removed from Freeware 8.0 distribution.

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 DCL Commands

The following release notes pertain to DCL commands.

3.2.1 CREATE/MAILBOX: Temporary Restriction

V8.2

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

3.2.2 DIAGNOSE Command No Longer Supported

V8.2

3.3 DECmigrate Not on Open Source Tools CD

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.

3.4 HP Secure Web Browser

The following notes pertain to the HP Secure Web Browser.

3.4.1 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 256 MB; however, 512 MB is highly recommended for more robust performance.

3.4.2 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.

3.5 Documentation Corrections

The following sections describe corrections and additions to various manuals in the OpenVMS documentation set.

3.5.1 *HP OpenVMS System Analysis Tools Manual*

For changes and updates to the *HP OpenVMS System Analysis Tools Manual*, see Chapter 4.

3.5.2 *HP OpenVMS Programming Concepts Manual*

The following corrections pertain to the *HP OpenVMS Programming Concepts Manual*:

3.5.2.1 Saving System Dumps

V8.3

The following changes should be made to the paragraph in Section 31.2, "Writing a Privileged Routine (User-Written System Service)":

"As a protected image, your program does not have the entire operating system programming environment at its disposal. Unless a module has the prefix SYS$ or EXE$, you must avoid calling it from an inner mode. In particular, do not call LIB$GET_VM or LIB$RET_VM from an inner mode. You can call OpenVMS RMS routines from executive mode but not from kernel mode."

LIB$GET_VM should be LIB$FREE_VM. You cannot call these LIBRTL routines directly, and you cannot call any routines that might now or in the future call these routines indirectly. This includes other routines within LIBRTL and the user-mode C library, among other libraries.

3.5.3 *HP OpenVMS Debugger Manual*: Client/Server Interface Correction

The Version 8.2 *HP OpenVMS Debugger Manual* contains incorrect information for the PC client interface kit. This release note corrects the information in Sections 11.1 and 11.2, as follows:

The debug server runs on OpenVMS systems. The client is the user interface to the debugger, from which you enter debugger commands that are sent to the server. The server executes the commands and sends the results to the client for display. The client runs on Microsoft® Windows® 95, Windows 98, Windows NT®, Windows 2000, and Windows XP®.

The correct client kit for the above client platforms is:

```
[DEBUG_CLIENTS011.KIT] DEBUGX86011.EXE
```

There is no special installation procedure for the components that run on OpenVMS. The system administrator should move the OpenVMS debug client kit listed above from the OpenVMS distribution media to a place accessible to PC users, such as a PATHWORKS share or FTP server. The client kit is a self-extracting .EXE file.

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.

By default, the debug client is installed in the `\Program Files\OpenVMS` Debugger folder. You can also click Browse to select an alternate location.

The installation creates an OpenVMS Debugger program folder that contains shortcuts to the following items:

- Debug client
- Debug client Help file
- `README` file
- Uninstall procedure

3.6 *HP OpenVMS DELTA/XDELTA Debugger Manual* Update

V8.3

The HP DELTA debugger was made available on OpenVMS I64 Version 8.2-1. The *HP OpenVMS DELTA/XDELTA Debugger Manual* has been revised in this release to include information about using DELTA on OpenVMS I64 systems.

3.7 *HP OpenVMS Utility Routines Manual* Updated

V8.3

The *HP OpenVMS Utility Routines Manual* has been revised to include the following information:

- A new chapter on the Traceback facility and TBK$ routines for I64 and Alpha systems
- Revisions to the Library utility chapter to include new and changed LBR$ routines.
- A new chapter on AES encryption for I64.

3.7.1 *HP OpenVMS I/O User's Reference Manual*: PTD$READ Clarification

The *HP OpenVMS I/O User's Reference Manual* does not fully describe the behavior of the read request. The following paragraph should replace the first paragraph of Section 6.5.1:

To read data from the pseudoterminal, the control program uses the PTD$READ routine. When a PTD$READ routine is called, the operating system queues a read operation. The read operation completes when the pseudoterminal has characters to output. The read request queries TTDRIVER whether there is data found to be returned. If so, the resulting string of characters is returned. If a read request is issued and no data is available, the read request is queued and then completed at a later time. In this case, the routine always returns at least one character. The read request may complete even when there are no characters available to output. In this rare case when TTDRIVER indicates that there is no

more data to be output and there is really no data, the read operation completes with zero bytes of data.

The following paragraphs replace Section D.4.4:

The PTD$READ routine reads data from the pseudoterminal. The read request completes with a minimum of one character and a maximum of the number of characters specified by the **readbuf_len** argument.

When a PTD$READ routine is called, the operating system queues a read operation. The read operation completes when the pseudoterminal has characters to output. The read request queries TTDRIVER whether there is data found to be returned. If so, the resulting string of characters is returned. If a read request is issued and no data is available, the read request is queued and then completed at a later time. In this case, the routine always returns at least one character. The read request may complete even when there are no characters available to output. In this rare case when TTDRIVER indicates that there is no more data to be output and there is really no data, the read operation completes with zero bytes of data.

3.7.1.1 PTD$ Pseudoterminal Driver Correction

V8.3

In references to the pseudoterminal driver control-connection routines, the PDT$ prefix is incorrect. The prefix should be PTD$.

3.7.2 *HP OpenVMS Version 8.2 New Features and Documentation Overview*: Librarian Utility Corrections

The following release notes provide corrected information about the OpenVMS I64 Librarian utility.

3.7.2.1 /REMOVE Qualifier Correction

In Section 4.8.2.3 of the *HP OpenVMS Version 8.2 New Features and Documentation Overview*, the description of the enhanced library /REMOVE qualifier is incorrect. The correct information follows.

The /REMOVE qualifier has been enhanced for the I64 Librarian utility. The format now allows you to specify the module instance of the symbol to be removed. The enhanced /REMOVE qualifier requests that the LIBRARY command delete one or more entries from the global symbol table of an object library.

3.7.2.2 Accessing ELF Object Libraries Correction

Section 4.8.3.2 of the *HP OpenVMS Version 8.2 New Features and Documentation Overview* contains incorrect information. The following text replaces information in that section:

Accessing ELF Object Libraries

ELF object modules are inherently random access modules, whereas OpenVMS Alpha objects, text modules, and so on, are sequential. To allow random access, a new library routine was created to map the ELF object modules into process P2 space so that applications can make random access queries. To recover virtual address space from this mapping, another library routine was created to remove this mapping. These new routines (LBR$MAP_MODULE and LBR$UNMAP_MODULE) work only with ELF object libraries. These entry points are 64-bit interfaces because they refer to P2 space.

Because of the random-access nature of ELF object files, the following operations are not allowed on ELF object libraries:

LBR$GET_RECORD
LBR$SET_LOCATE
LBR$SET_MOVE

Because inserting modules into the library is a sequential operation, LBR$PUT_RECORD is allowed on ELF object libraries. Because the ELF object modules are not segmented into records, you need to provide the module's on-disk size when calling LBR$PUT_MODULE or upon the first call to LBR$PUT_RECORD when writing a module into the library.

The C code fragment in the following example illustrates how to use LBR$PUT_RECORD to insert an object module:

```
bufdesc->dsc$a_pointer = &p0_buffer ;
bytes_to_transfer = module_size ;

while ( bytes_to_transfer )  {
   transfer = MIN ( bytes_to_transfer ,
                    ELBR$C_MAXRECSIZ ) ;

   bufdesc->dsc$w_length = transfer ;

   status = lbr$put_record (  library_index ,
                            & bufdesc ,
                            & txtrfa ,
                              module_size ) ;
   if ( (status & 1) == 0 )
      break ;

   bytes_to_transfer      -= transfer ;
   bufdesc->dsc$a_pointer += transfer ;
   } ;

if ( (status & 1) == 1 )
   status = lbr$put_end ( library_index ) ;
```

To avoid making several calls to LBR$PUT_RECORD, a new library routine, LBR$PUT_MODULE, has been created.

3.7.3 *HP OpenVMS RTL Library (LIB$) Manual* Corrections

V8.2-1

The following sections provide additions and corrections to Version 8.2 of the *HP OpenVMS RTL Library (LIB$) Manual*.

3.7.3.1 *HP OpenVMS RTL Library (LIB$) Manual*: Clarification of Rounding Rule for LIB$CVT_DX_DX

V8.2-1

In the description of the LIB$CVT_DX_DX routine in the *HP OpenVMS RTL Library (LIB$) Manual*, the following paragraph under ''Guidelines for Using LIB$CVT_DX_DX'' should contain specific information about the rounding rule that is used:

Results are always rounded instead of truncated, except for the case described below. Note that loss of precision or range may be inherent in the destination data type or in the NBDS destination size. No errors are reported if there is a loss of precision or range as a result of destination data type.

This paragraph should be modified as follows:

Results are always rounded instead of truncated, except for when the source and destination are both NBDS and no scaling is requested. That case is described more fully in a later rule. LIB$CVT_DX_DX uses the VAX_ROUNDING rule. Note that loss of precision or range may be inherent in the destination data type or in the NBDS destination size. No errors are reported if there is a loss of precision or range as a result of destination data type. For details about the VAX_ROUNDING rule, refer to the description of CVT$CONVERT_FLOAT.

3.7.4 *HP OpenVMS RTL Library (LIB$) Manual*: Clarification of Platform Restrictions

V8.2–1

The *HP OpenVMS RTL Library (LIB$) Manual* incorrectly identifies the following routines as being available on both Alpha and I64. These routines are available only on Alpha:

- LIB$GET_CURR_INVO_CONTEXT
- LIB$GET_INVO_CONTEXT
- LIB$GET_INVO_HANDLE
- LIB$GET_PREV_INVO_CONTEXT
- LIB$GET_PREV_INVO_HANDLE
- LIB$PUT_INVO_REGISTERS

The *HP OpenVMS RTL Library (LIB$) Manual* also should specify that the LIB$GET_UIB_INFO routine is available only on I64.

The routines relating to invocation contexts and invocation handles that are I64 only include the following:

- LIB$I64_CREATE_INVO_CONTEXT
- LIB$I64_FREE_INVO_CONTEXT
- LIB$I64_GET_CURR_INVO_CONTEXT
- LIB$I64_GET_CURR_INVO_HANDLE
- LIB$I64_GET_INVO_CONTEXT
- LIB$I64_GET_INVO_HANDLE
- LIB$I64_GET_PREV_INVO_CONTEXT
- LIB$I64_GET_PREV_INVO_HANDLE

For additional information about these routines, refer to the *HP OpenVMS Calling Standard*.

3.7.5 *HP OpenVMS System Manager's Manual*: IPC Commands Restriction

V8.2-1

Section 9.15, Using Interrupt Priority Level C (IPC), in the *HP OpenVMS System Manager's Manual, Volume 1: Essentials* incorrectly states that you can use IPC commands on I64 and all Alpha systems. This is not correct. The documentation has been changed to include the following statement:

```
For OpenVMS Versions 8.2 and 8.2--1, you cannot use IPC commands
on I64 systems or on ES47 or GS1280 Alpha systems if you booted
from a Graphic console.
```

3.7.5.1 $PUTMSG System Service Correction

V8.2-1

The $PUTMSG prototype is incorrectly documented in Version 8.2 of the *HP OpenVMS System Services Reference Manual*. The correct prototype is as follows:

C Prototype

```
int sys$putmsg (void *msgvec, int (*actrtn)(__unknown_params),
void *facnam, unsigned __int64 actprm);
```

Note that the return value from **actrtn* is indeed checked to determine whether or not the message is input.

The documentation source file has been corrected, and the correction will appear in the next version of the *HP OpenVMS System Services Reference Manual* and in online help.

3.7.6 *HP Volume Shadowing for OpenVMS*: Corrections to Memory Requirements

V8.2–1

Starting with OpenVMS Version 7.3, additional memory is required to run Volume Shadowing for OpenVMS, as described in the Version 7.3–2 *HP Volume Shadowing for OpenVMS* manual in Chapter 1 under Hardware Environment, Memory Requirements.. Note the following corrections:

- In the sentence, "For example, for a shadow set with 200 GB of storage per member, 420 KB of memory is required for its write bitmaps for every node in the cluster.", the total of 420 should be 400.
- Three paragraphs later, in the sentence, "For example, a system with 10 shadow sets mounted, with each shadow set consisting of 50-GB member disks, would require an additional 1,119 KB of memory.", the total of 1,119 should be 1,069, as it appears in the calculation that follows on the same page.

These totals will be corrected in the next revision of *HP Volume Shadowing for OpenVMS*.

3.8 Network Update Restrictions from Version 8.2 to Version 8.2–1

V8.2–1

OpenVMS Version 8.2–1 supports network update of the operating system from Version 8.2 to Version 8.2–1. Network update is supported only over the core I/O LAN cards on systems supported by OpenVMS Version 8.2. Refer to the *HP OpenVMS Version 8.2–1 for Integrity Servers Upgrade and Installation Manual* for more information.

In addition, there is also a hardware configuration restriction for network booting. Unlike Alpha consoles, where the speed and duplex setting for the network adapter can be selected at the console, the Integrity server console and network boot drivers perform autonegotiation only. The network switch nearest the Integrity server boot client must be set to autonegotiate for a successful network boot. Failure to set the switch to autonegotiate results in an inability to complete the network boot process.

3.9 Synchronous Data Links Not Supported

V8.2–1

OpenVMS Version 8.2–1 does not support any synchronous data link hardware on Integrity servers.

3.10 Duplex-Mode Mismatch Errors Reported by the LAN Drivers

V8.3

A duplex-mode mismatch condition occurs when a LAN device is operating in full-duplex mode and the other end of the cable, typically a switch port, is operating in half-duplex mode. The reverse is also true. A common network configuration error that results in a duplex mode mismatch condition occurs when the switch port is set to autonegotiate the speed and duplex settings, and the LAN device is set to a fixed setting of full duplex. In this configuration, autonegotiation by the switch results in the selection of half-duplex mode and the LAN device is set to full-duplex mode, and a duplex-mode mismatch occurs.

The consequence of a duplex-mode mismatch is typically a performance degradation. In addition, the IEEE 802.3 specification that describes the autonegotiation process suggests that a duplex-mode mismatch can result in data corruption. For most LAN devices, the only consequence of a duplex mode mismatch is the performance degradation. For some LAN devices, packet data is corrupted with good CRC, resulting in packet corruption undetected by the LAN subsystem. These devices include all Broadcom-based NICs and embedded LOM chips. On Alpha systems, these include the DEGPA, DEGXA, BCM5703 LOM on the AlphaServer DS25, and any implementations using the dual-port BCM5704 chip. On Integrity systems, these include the A6847A, A6725A, A9782A, A9784A, AB465A, and BCM5701 LOM on the rx2600; BCM5703 LOM on other systems; and the A6794A.

In prior versions of OpenVMS, the LAN drivers attempt to detect the duplex-mode mismatch condition. Once an hour while the condition exists, they issue a console message and error log message warning of the condition.

In OpenVMS Version 8.3, the frequency of the messages is increased from once per hour to once every 36 seconds for any Broadcom-based LAN devices. The frequency remains at once per hour for non-Broadcom-based LAN devices. In addition, to increase the visibility of these messages, the console messages are sent to OPCOM and to the LANACP log file (SYS$MANAGER:LAN$ACP.LOG).

The purpose of this note is to underscore the importance of avoiding duplex-mode mismatches, particularly when this condition results in undetected data corruption for Broadcom-based devices.

Note that the LAN drivers detect a duplex mode mismatch condition by monitoring device errors. The detection is not perfect, so the LAN drivers refer to the condition as a "potential duplex-mode mismatch." Upon noticing these messages, a system or network manager should inspect the LAN counters and LAN device settings to ensure a duplex-mode mismatch condition does not exist.

3.11 Failure of AUDIT_SERVER to Initiate During Boot

V8.3

System managers should be aware that, during boot, if the AUDIT_SERVER fails to initiate for any reason, the startup process enters a retry loop that attempts to restart the AUDIT_SERVER until the condition preventing the initiate to complete is cleared and AUDIT_SERVER initiates correctly. This behavior is deliberate and is designed to prevent the system from running in a compromised security state.

Conditions that can prevent complete initiation include the following:

- Invalid journal file specification in the audit server database
- Corrupt audit server database (`SYS$MANAGER:VMS$AUDIT_SERVER.DAT`)
- Corrupt object server database (`SYS$SYSTEM:.VMS$OBJECTS.DAT`)
- Failed access to target disk
- Failure to synchronize with other audit servers on the cluster
- Required resources being held by other audit servers on the cluster

Clearing the condition might require manual intervention. The action required depends on the fault source. Corrective action can include restarting the AUDIT_SERVER processes on other cluster nodes or rebooting the affected node in MINIMUM state and manually correcting the fault. Corrupt database files can be identified by renaming the files and then restarting the AUDIT_SERVER. The server recreates the absent files and populates them with system default entries.

For more information about booting options, see Chapter 4 of *HP OpenVMS System Manager's Manual, Volume 1: Essentials*.

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 8.3 New Features and Documentation Overview*.

4.1 Monitor Utility Changes

V8.3

The Monitor utility (MONITOR) has undergone several changes since OpenVMS Version 7.3-2. Most of these changes are related to providing improved formatting of the recording file and including additional class data. These changes have introduced some compatibility issues between data collected by one version of MONITOR that is subsequently processed by another version. This section discusses these issues.

4.1.1 Version-to-Version Compatibility of MONITOR Data

Because the body of data MONITOR collects can change at each release, it is not always possible to view MONITOR data collected in one version on a different version.

The level of compatibility between releases depends on whether you examine recorded binary data from a file (that is, playback) or live data from another cluster node. In general, playing back recorded data provides more compatibility than monitoring live remote data.

4.1.2 Playing Back Data from a Recording File

Each file of recorded MONITOR binary data is identified by a MONITOR recording file-structure level ID. You can see this ID by entering the DCL command DUMP /HEADER /PAGE on the file. The following table lists ome recent MONITOR versions and their associated structure level IDs:

Operating System Version	MONITOR Recording File Structure ID
OpenVMS Version 7.3-2 with remedial kit[1]	MON32050
OpenVMS Versions 8.2, 8.2-1 with remedial kit[1]	MON01060

[1]These remedial kits are proposed kits that might be issued for the sole purpose of providing improved compatibility.

Usually, for you to be able to play back a single MONITOR recording file, the last two digits of the structure level ID must match those of the running MONITOR version. For example, if you are running OpenVMS Version 7.3-2, you can play back a file from Version 7.3-2 but not one from Version 8.2.

However, MONITOR Versions 8.2 and higher are specially coded to read recording files with structure level IDs ending in "50." In addition, a utility in SYS$EXAMPLES, called MONITOR_CONVERT.C, converts a MONxx060 file to a MON31050 file. This allows the resulting file to be read by versions prior to Version 8.2. See MONITOR_CONVERT.C for instructions for building and running the program.

Note that, even though you are allowed to play back a file, certain MONITOR data classes within the file might not be available. This can happen if you are using an older MONITOR version to play back a file created by a newer MONITOR version.

Finally, note that, when you produce a multifile summary from several recording files, all 8 characters of the structure level ID from all the files must match.

4.1.3 Monitoring Live Remote Data across a VMScluster

V8.3

In addition to the recording file structure level ID, each MONITOR version also has an associated ‘‘server version number.’’ The server version number identifies the version of MONITOR data to make it possible to serve live data from one node to another in an OpenVMS Cluster. For you to monitor data from another cluster node, both the monitoring node and the monitored node must have the same server version number. If the versions are not the same, the following error message is displayed:

```
%MONITOR-E-SRVMISMATCH, MONITOR server on remote node is an
                        incompatible version
```

Some recent MONITOR versions and their associated server version numbers are in the following table:

Operating System Version	MONITOR Server Version Number
OpenVMS Version 7.3-2	5
OpenVMS Version 7.3-2 with remedial kit[1]	7
OpenVMS Version 8.2, 8.2-1 with remedial kit[1]	8

[1]These remedial kits are proposed kits, which might be issued for the sole purpose of providing improved compatibility.

If cross-node live monitoring is not possible because incompatibility of versions, you might be able to view the statistics you want through a combination of recording (to a file) and playback.

4.2 Updated Recommended File Security Attributes

V8.3

The following table lists updated recommended protection profiles for files listed in the *HP OpenVMS Guide to System Security*. (The file VMS$PASSWORD_HISTORY.DATA is omitted from the current version of the manual but will be included in the next revision.)

File	Protection
RIGHTSLIST.DAT	S:RWE,O:RWE,G,W
SYSUAF.DAT	S:RWE,O:RWE,G,W
VMS$OBJECTS.DAT	S:RWE,O:RWE,G:RE,W
VMS$PASSWORD_HISTORY.DATA	S:RWE,O:RWE,G,W

The file owner should be a UIC with a group within the system range (less than MAXSYSGROUP system parameter). Values of [1,1] or [SYSTEM] (1,4) are recommended.

4.3 System Management Notes

The following sections describe updates to system management and maintenance.

4.3.1 CIMSERVER Process Recommendation (I64 Only)

V8.3

For optimal performance, HP recommends that the account running the CIMSERVER process (usually the SYSTEM account) have a PGFLQUOTA of at least 1 GB (PGFLQUOTA = 2000000).

This restriction will be lifted with the next release of WBEM Services for OpenVMS.

4.4 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.

4.5 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.

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

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 and higher. Version 5.5 is backward compatible with OpenVMS Version 6.2 and higher.

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

The ECP Analyzer currently is 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 using this utility. Certain privileged operations within this utility could render OpenVMS I64 unbootable.

4.10 EFI Shell Precautions on Shared or Shadowed System Disks

V8.2-1

On each Integrity system disk, there can exist up to two FAT partitions that contain OpenVMS boot loaders, EFI applications and hardware diagnostics. The OpenVMS bootstrap partition and, when present, the diagnostics partition are respectively mapped to the following container files on the OpenVMS system disk:

SYS$LOADABLE_IMAGES:SYS$EFI.SYS
SYS$MAINTENANCE:SYS$DIAGNOSTICS.SYS

The contents of these FAT partitions appear as fs*n*: devices at the console EFI Shell> prompt. These fs*n*: devices can be directly modified by user command input at EFI Shell> prompt, and by EFI console or EFI diagnostic applications. Neither OpenVMS nor any EFI console environments that might share the system disk are notified of partition modifications; OpenVMS and console environments are entirely unaware of these console modifications. Accordingly, you must ensure the proper coordination and proper synchronization of the changes with OpenVMS and with any other EFI consoles that might be in use.

You must take precautions when modifying the console in configurations using either or both of the following:

- OpenVMS host-based volume shadowing for the OpenVMS I64 system disk
- Shared system disks and parallel EFI console access across Integrity environments sharing a common system disk

You must preemptively reduce these OpenVMS system disk environments to a single-member host-based volume shadowset or to a non-shadowed system disk, and you must externally coordinate access to avoid parallel accesses to the Shell> prompt whenever making shell-level modifications to the fs*n*: devices, such as:

- Installing or operating diagnostics within the diagnostics partition
- Allowing diagnostics in the partition (or running from removable media) to modify the boot or the diagnostic partition on such an OpenVMS I64 system disk
- Otherwise directly or indirectly modifying the boot or the diagnostics partition within these environments from the Shell> prompt

If you do not take these precautions, any modifications made within the fs*n* device associated with the boot partition or the device associated with the diagnostic partition can be overwritten and lost immediately or after the next OpenVMS host-based volume shadowing full-merge operation.

For example, when the system disk is shadowed and changes are made by the EFI console shell to the contents of these container files on one of the physical members, the volume shadowing software has no knowledge that a write was done to a physical device. If the system disk is a multiple member shadow set, you must make the same changes to all of the other physical devices that are the current shadow set members. If this is not done, when a full merge operation is next performed on that system disk, the contents of these files might regress. The merge operation might occur many days or weeks after any EFI changes are done.

Furthermore, if a full merge is active on the shadowed system disk, you should not make changes to either file using the console EFI shell.

To suspend a full merge that is in progress or to determine the membership of a shadow set, see the HBMM chapter of the *HP OpenVMS Version 8.2 New Features and Documentation Overview*.

These precautions apply only to Integrity system disks that are configured for host-based volume shadowing, or that are configured and shared across multiple OpenVMS I64 systems. Configurations that are using controller-based RAID, that are not using host-based shadowing with the system disk, or that are not shared with other OpenVMS I64 systems, are not affected.

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, see the *HP OpenVMS Guide to System Security*. Also see Section 2.13.1 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 OpenVMS Cluster Systems

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

4.13.1 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.

4.13.2 Temporary Exceptions

V8.2

The following exceptions are temporary:

- 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.
- I64 Satellite systems that use device naming (also known as port allocation classes) require an additional step to operate correctly in this release. On the satellite boot server node, edit the file device:

  ```
  [SYSn.SYSCOMMON.SYS$LDR]SYS$MEMORYDISK.DAT
  ```

  where where *device* is the disk that contains the satellite's root and where *n* is the root of the satellite system) and add the following line to the file:

  ```
  SYS$SYSTEM:SYS$DEVICES.DAT, text
  ```

  You can safely ignore the "Do Not Edit" comment at the top of the file in this case. The list of files in SYS$MEMORYDISK.DAT is not order-dependent. This problem is expected to be resolved for the final release.

4.13.3 Satellite Booting and LAN Failover

V8.3

For Alpha and Integrity cluster satellites, the network boot device cannot be a prospective member of a LAN Failover Set. For example, if you create a LAN Failover Set, LLA consisting of EWA and EWB, to be active when the system boots, you cannot boot the system as a satellite over the LAN devices EWA or EWB.

4.13.4 Creation of Error Log Dump File for Integrity Server Satellite Systems

V8.3

Integrity server satellite systems require DOSD (Dump Off the System Disk) for both the system crash dump file and system error log. Autogen will create the system dump on an appropriate disk once DOSD is enabled, but does not attempt to create the error log dump file (SYS$ERRORLOG.DMP). In order to preserve error-log entries across system failure, the error log dump file must be created by hand.

See the *HP OpenVMS System Manager's Manual, Volume 2: Tuning, Monitoring, and Complex Systems* for information on enabling DOSD. After running AUTOGEN to create the dump file on the appropriate device, create the error log dump as follows:

1. Use SYSGEN to obtain the values of the the system parameters ERRORLOGBUF_S2 and ERLBUFFERPAG_S2.
2. Calculate (ERRORLOGBUF_S2 * ERLBUFFERPAG_S2) + 10.
3. Use the `SYSGEN CREATE` command to create the file as follows:

   ```
   SYSGEN> CREATE dev:[SYSn.SYSEXE]SYS$ERRLOG.DMP/SIZE=filesize
   ```

   where

   ```
   dev = a device in the DOSD list
   n = the system root for the satellite
   filesize = the value calculated in step 2.
   ```

   HP anticipates that AUTOGEN will be enhanced in the future to perform this operation.

4.14 Cluster Compatibility Patch Kits

V8.3

Before introducing an OpenVMS Version 8.2–1 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. Note that these kits are version specific.

The versions listed in Table 4–2 are supported in a warranted configuration. For more information about these configurations, see the *HP OpenVMS Version 8.2–1 for Integrity Servers Upgrade and Installation Manual*.

Table 4–2 lists the facilities that require patch kits and the patch kit file names. Each patch kit has a corresponding readme file by the same name with a .README file extension.

You can either download the patch kits from the following Web site or contact your HP Support representative to receive the patch kits on media appropriate for your system:

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

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 4–2 Patch Kits Required for Cluster Compatibility

Facility	Patch Kit File Name
OpenVMS Alpha Version 7.3-2	
Update kit with most patch kits except those listed in this section	VMS732_UPDATE-V0600.
C RTL	VMS732_ACRTL-V0100
Drivers	VMS732_DRIVER-V0200
PCSI	VMS732_PCSI-V0100
OpenVMS Alpha Version 8.2	
	VMS82A_UPDATE-V0200
DECnet-Plus for OpenVMS Alpha ECO1	DNVOSIECO01_V82[1]
OpenVMS I64 Version 8.2	
	VMS82I_UPDATE-V0200

[1]This kit is required if you are running this software in your configuration.

4.14.1 Patch Kits Needed for Cluster Compatibility

V8.2

Before introducing an OpenVMS Version 8.2 (or higher) 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 4–2 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 8.3 Upgrade and Installation Manual*.

Table 4–2 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 4–2 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-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 4.13.2.

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 8.3 Upgrade and Installation Manual*.

4.14.2 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 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*.

4.14.3 DDT Intercept Establisher Routines and Device Configuration Notification Results

V8.3

To ensure proper behavior of certain routines, a patch kit is required. Using those routines without the required patch kit can result in system hangs, crashes, or data corruption, and is not supported by HP.

For more information about these routines, see the *HP OpenVMS Alpha Version 7.3–2 New Features and Documentation Overview* manual.

4.14.4 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.

4.14.5 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.

4.14.6 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.

4.14.7 Availability Manager AVAIL_MAN_BASE Kit Fails to Collect Lock Contention Information

The OpenVMS Version 8.3 installation includes the installation of the AVAIL_MAN_BASE SIP kit. This kit, in turn, installs the Availability Manager Data Collector.

The Data Collector in the AVAIL_MAN_BASE SIP kit has a defect that causes it to collect lock contention data incorrectly. To correct this problem, install the AVAIL_MAN_COL Version 2.6 kit, which is available on the operating system update media.

4.15 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.

4.15.1 Galaxy Definitions

V8.2

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

Table 4–3 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.

4.16 Multiple nPartitions on Cell-based Systems

V8.2-1

If you have multiple nPartitions on your HP Integrity rx7620, HP Integrity rx8620, or HP Integrity Superdome servers, and you are running a multi-operating system environment with OpenVMS on one of the nPartitions, one of the other operating systems might register an error or event on the System Event Log (SEL) while OpenVMS is booting. OpenVMS holds the SEL until it has produced a table of Field Replaceable Units (FRU), which might cause other operating systems to register an error or an event.

4.16.1 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.

4.16.2 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.

4.16.3 Galaxy on ES40: Turning Off Fast Path

Permanent Restriction

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.

4.17 OpenVMS Management Station

V8.2

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

The OpenVMS installation includes OpenVMS Management Station Version 3.2D.

4.18 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.

4.19 System Parameters

The following sections contain notes related to system parameters.

4.19.1 New System Parameters

V8.3

To learn about new system parameters, see the *HP OpenVMS Version 8.3 New Features and Documentation Overview*.

4.19.2 Obsolete System Parameters

V8.3

The following system parameters are marked as obsolete in OpenVMS Version 8.3:

- SMP_CPUS
- SMP_CPUSH
- IO_PREFER_CPU
- IO_PREFER_CPUS
- NPAG_AGGRESSIVE
- NPAG_GENTLE
- SCH_CTLFLAGS
- SHADOW_REC_DLY
- TTY_SILOTIME
- BALSETCNT
- BREAKPOINTS
- MMG_CTLFLAGS
- MULTITHREAD
- NISCS_MAX_PKTSZ
- NISCS_PORT_SERV
- SECURITY POLICY
- SHADOW_HBMM_RTC
- SHADOW_PSM_RDLY
- SHADOW_SYS_DISK
- WBM_MSG_UPPER

The following new parameters replace the preceding ones:

- SMP_CPU_BITMAP
- IO_PRCPU_BITMAP

For more information about these new system parameters, see the *HP OpenVMS System Management Utilities Reference Manual* or online help.

4.19.3 System Parameter Changes

V8.3

The following system parameters are changed in OpenVMS Version 8.3. For more information, see the *HP OpenVMS System Management Utilities Reference Manual*.

- BALSETCNT - wording changes
- BREAKPOINTS - now dynamic
- MMG_CTLFLAGS - additional bits defined; wording changes
- MULTITHREAD - I64 support added
- NISCS_MAX_PKTSZ - wording changes
- NISCS_PORT_SERV - bit definition changes
- SECURITY POLICY - Bits 13 and 14 defined
- SHADOW_HBMM_RTC - wording changes
- SHADOW_PSM_RDLY - spelling corrected (from SHADOW_PSM_DLY)
- SHADOW_SYS_DISK - wording changes
- WBM_MSG_UPPER - default changed

For detailed descriptions of these parameters see the online help or the *HP OpenVMS System Management Utilities Reference Manual*.

4.19.4 Documentation Error: LCKMGR_CPUID System Parameter

V8.3

The *OpenVMS Performance Management* manual contains several references to the system parameter LCKMGR_CPUID as LOCKMGR_CPU. This latter reference is incorrect and will be corrected the next time the manual is updated.

4.19.5 MMG_CTLFLAGS: Documentation Error

V8.2

There is an error in the description of Bit 1 of the MMG_CTLFLAGS system parameter 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."

4.20 Terminal Fallback Facility (TFF)

V8.2

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 4–4 describes these additional tables.

Table 4–4 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.

4.21 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.

4.22 Recommended Caching Methods

Permanent Restriction

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.

4.23 Volume Shadowing for OpenVMS

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

4.23.1 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
```

4.23.2 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.

4.23.3 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:.)

4.23.4 KZPDC (Smart Array 5300) Restrictions

Permanent Restriction

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.

4.23.5 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.

4.23.6 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.

4.24 Authentication and Credential Management Extension (ACME)

4.24.1 New and Changed Features

V8.3

Following are new and changed features in Version 8.3.

- ACME_SERVER restart changes

  In previous releases, the ACME_SERVER process recorded all configuration commands in a staging file, SYS$SYSTEM:ACME$SERVER_CONFIG.TMP. Upon restart, the server used this file to create a data file, SYS$SYSTEM:ACME$SERVER_RESTART.DAT, which contained selected configuration information from the staging file to be processed during restart.

  This method of restoring the server's configuration state following a restart is no longer used and is replaced by a site-specific startup procedure, SYS$MANAGER:ACME$START.COM. The executive-mode logical name ACME$START is used to locate this file.

  SYS$MANAGER:ACME$START.COM is run as a result of one of the following conditions:

  - SET SERVER ACME/START=AUTO command is issued.
  - SET SERVER ACME/RESTART command is issued.
  - Unexpected condition causes an automatic server restart.

  The SYS$MANAGER:ACME$START.LOG file contains any information produced during restart in the event of problems.

  Users can modify the SYS$MANAGER:ACME$START.COM file to define which agents are configured during a restart as well as any other server configuration options. This file is not replaced during system upgrades. The SYS$MANAGER:ACME$START.TEMPLATE file contains the HP version of the file.

- New SET SERVER ACME/START[=AUTO] keyword

  The optional keyword AUTO causes the server to startup and configure itself using the SYS$MANAGER:ACME$START.COM procedure. By default, the server starts with only the OpenVMS ACME agent configured.

- SET SERVER ACME/CONFIG=THREAD_MAX ignored on Integrity servers

  The SET SERVER ACME/CONFIG=THREAD_MAX command is ignored on Integrity servers for this release. Only one worker thread is active.

- Optional kit for SYS$ACM-enabled LOUGINOUT.EXE and SETP0.EXE Images

SYS$ACM-enabled LOGINOUT.EXE and SETP0.EXE (SET PASSWORD) images are available as an optional installation kit. When used with non-OpenVMS ACME agents, these images provide additional authentication features for sites that require them.

For information about the operational requirements for this kit and a list of supported ACME agents, please refer to SYS$HELP:ACME_DEV_README.TXT.

4.25 Default Startup Order Required for OpenVMS and Kerberos ACME Agents

V8.3

The default settings for starting the OpenVMS ACME agents allow logins to function normally. If the default order of startup for the OpenVMS and Kerberos ACME agents is changed so that the Kerberos ACME is started first, accounts that are not in the Kerberos realm might not be able to log in. Changing the default order of startup is not supported in this release of OpenVMS.

If the startup order is changed, you can change it back to the default order by performing the following procedure.

Edit SYS$MANAGER:ACME$START.COM to search for the section (near the end of the command procedure) where you can specify the desired agent ordering. Change the last line (beginning with AGENT_LIST) so that it appears in the procedure.

```
$! A specific agent ordering can be specified in AGENT_LIST.
$!
$! If the list is empty, the agents will be enabled in the order that
$! they were configured. Some agent startup procedures may alter
$! the agent order. You can override that ordering here. Consult the
$! agent documentation you are using to ensure that the ordering you
$! specify is supported by that agent.
$!
$! For example
$!
$!      AGENT_LIST = "VMS,MSV1_0"
$!
$! will enable the VMS and MSV1_0 agents (and only those agents) in
$! that order.
$!
$ AGENT_LIST = "VMS,ACME_KRB_DOI"
```

4.26 Traceback API Problem Fixed

In OpenVMS I64 Version 8.2, an error occurred when the *pc_rel* (relative PC value) or *image_desc* (image name string descriptor) arguments were specified as zero. The Traceback facility now correctly ignores these arguments and continues Traceback processing.

4.27 WBEM Services for OpenVMS Version 2.0 Release Notes

The following release notes address late-breaking information about Version 2.0 of WBEM Services for OpenVMS.

4.27.1 Based on OpenPegasus 2.5

WBEM Services for OpenVMS Version 2.0 is based on the OpenPegasus 2.5 code stream of the The Open Group's Pegasus open source project.

4.27.2 Supports nPartitions and iCAP

Version 2.0 supports local nPartitions and iCAP providers. Only the functions and capabilities needed by these providers are supported.

4.27.3 Not Designed for Homogeneous Clusters

Pegasus, a UNIX-based product, is not designed for homogeneous clusters. To have the cimserver run on more than one node on a cluster common system disk, install the product on separate roots so that the repository, trace, and log file directories do not conflict.

4.27.4 Restart cimserver.exe to Unload Providers on OpenVMS

After entering the cimprovider *-r* command, you must stop and restart the cimserver to complete the process of replacing a provider. (OpenVMS does not support unloading a dynamically loaded image.)

4.27.5 Use Quotes Around Command Line Options

Be sure to use quotes around a command line option to preserve its case. For example,
Correct:

```
$ cimmofl "-E" "--xml"
```

Incorrect:

```
$ cimmof -E -xml
```

4.27.6 Clean Up WBEMCIM Repository Directory Tree After Uninstall

The DCL command PRODUCT REMOVE WBEMCIM does not remove the repository directory tree that WBEM_SERVICES$SETUP.EXE creates after installation. You need to delete SYS$COMMON:[WBEM_SERVICES.VAR...]*.*;* by hand.

If you do not delete the directory tree, when you upgrade or reinstall WBEM, multiple "File already exists" errors are reported when you run WBEM_ SERVICES$SETUP.EXE.

4.28 Missing $SIGNAL_ARRAY_64 System Service on I64

V8.3

The system service $SIGNAL_ARRAY_64 is missing on I64. Whe you call this service on I64, you receive a return status of `SS$_NOT_LOADED (4026 decimal)`. This service will be added in a future release of OpenVMS on I64.

To work around this problem, you can find the address of the 64-bit signal array in the mechanism array at `offset chf$ph_mch_sig64_addr`. For example, in C, you can replace the following statement:

```
status = sys$signal_array_64(mech_array, &sig64_array);
```

with the following statement:

```
sig64_array = ((CHFDEF2 *)mech_array)->chf$ph_mch_sig64_addr;
```

4.29 *HP OpenVMS System Analysis Tools Manual*

V8.3

The following corrections pertain to the *HP OpenVMS System Analysis Tools Manual*. This manual was not updated for OpenVMS Version 8.3, but the corrections noted here and the additions described in the *HP OpenVMS Version 8.3 New Features and Documentation Overview* are included in online help for the SDA utility and in related commands for ANALYZE and System Services logging:

4.29.1 Definitions of Bits in DUMPSTYLE

The following row should replace the last row in Table 2-1:

5	32	0= Write all processes and global pages in a selective dump. 1= 1= Only write key processes and global pages in a selective dump. This bit is ignored when writing a full dump (bit 0 = 0). This bit should be set only if the priority processes have been correctly set up, as described in *HP OpenVMS System Manager's Manual, Volume 2: Tuning, Monitoring, and Complex Systems*.

4.29.2 Saving System Dumps

The following paragraphs should be appended to Section 2.2.2:
When a dump is being analyzed, it is useful to have data available that cannot be written to the dump file at the time of the system crash. This data includes the full file specification associated with a file identification and, on I64 systems, the unwind data for images activated in processes.

If the dump is being analyzed on the system where it was originally written, this data can be collected for use in the current SDA session using the COLLECT command. If the dump is being copied for analysis elsewhere, the COPY/COLLECT command may be used to collect the data and append it to the copy being written. If the COPY/COLLECT command is used after a COLLECT command, the data already collected is appended to the dump copy.

By default, a copy of the original dump, as written at the time of the system crash, includes collection. You can use COPY/NOCOLLECT to override this. Conversely, a copy of a dump previously copied by SDA without collection (COPY/NOCOLLECT) does not include collection. You can use COPY/COLLECT to override this.

Copying a dump that already contains an appended collection always includes that collection.

For all file and unwind data to be collected successfully, all disks that were mounted at the time of the system crash should be remounted and accessible to the process running SDA. If SDA is started early during startup to save the contents of the dump (for example, using CLUE$SITE_PROC; see Section 2.2.3), but disks are not mounted until a batch job is run, then the COPY/NOCOLLECT command should be used in the CLUE$SITE_PROC command procedure. Once all disks are mounted, a COPY/COLLECT can be used to save file and/or unwind data.

If the COPY and the COLLECT operations cannot be done as a single step, entering the COLLECT/SAVE command writes the collection to a separate file that can be used later in conjunction with the dump file. A later copy combines the two files.

4.29.3 Invoking SDA When Rebooting the System

In Section 2.2.3, the following should be appended to the end of the paragraph on the CLUE HISTORY command:
You might need to include the /NOCOLLECT qualifier to the COPY command. For more information, see the preceding section for details.

4.29.4 SDA CONTEXT

In Section 2.5, in the first list of the SET PROCESS and SHOW PROCESS commands, the SHOW PROCESS/SYSTEM and SHOW PROCESS/NEXT lines should be juxtaposed, and the following lines should be added to the list:

```
VALIDATE PROCESS/POOL process-name
VALIDATE PROCESS/POOL/ADDRESS=pcb-address
VALIDATE PROCESS/POOL/INDEX=nn
VALIDATE PROCESS/POOL/NEXT
VALIDATE PROCESS/POOL/SYSTEM
```

At the end of the section, the following lines should be added to the bottom of the list of SET PROCESS and SHOW PROCESS commands:

```
VALIDATE PROCESS/POOL process-name
VALIDATE PROCESS/POOL/ADDRESS=pcb-address
VALIDATE PROCESS/POOL/INDEX=nn
VALIDATE PROCESS/POOL/NEXT
```

4.29.5 SDA Symbol Initiation

In Section 2.6.1.4, the following line should immediately precede the Section 4.29.5 heading:

Symbols can include lowercase letters. Commands that manipulate symbols (such as DEFINE, SHOW SYMBOL, and UNDEFINE) require such symbols to be within quotes.

4.29.6 ANALYZE

In the Parameters section, Chapter 3, the of the file specifcation description should be as follows:

Name of file that contains the dump you want to analyze. If no file specification is given with an ANALYZE/CRASH_DUMP command, the default is the highest version of SYS$SYSTEM:SYSDUMP.DMP. If this file does not exist, SDA prompts you for a file name. If any field of file specification is given, the remaining fields default to the highest version of SYSDUMP.DMP in your default directory.

The following should be added to the Parameters section:

Collection-file-name

Name of the file to be used by SDA that contains the file ID translation data and/or unwind data.

4.29.7 Crash_Dump

In Chapter 3, in the Format section, `filespec` should be `[ filespec ]` (that is, the *filespec* parameter is now optional).

4.29.8 SDA Commands

In the the Introduction to Chapter 4, FLT should be removed and the following commands added:

COLLECT
SHOW EFI
SHOW VHPT
VALIDATE POOL
VALIDATE PROCESS

4.29.9 Copy Command

In Chapter 4, the following content should be appended to the COPY command description:

When a dump is being analyzed, it is useful to have data available that cannot be written to the dump file at the time of the system crash. This data includes the full file specification associated with a file identification, and, on OpenVMS I64, the unwind data for images activated in processes.

If the dump is being analyzed on the system where it was originally written, this data can be collected for use in the current SDA session using the `COLLECT` command. If the dump is being copied for analysis elsewhere, the `COPY/COLLECT` command can be used to collect the data and append it to the copy being written. If the `COPY/COLLECT` command is used after a `COLLECT` command, the data already collected is appended to the dump copy.

By default, a copy of the original dump, as written at the time of the system crash, will include collection. You can use `COPY/NOCOLLECT` to override this. Conversely, a copy of a dump previously copied by SDA without collection `(COPY/NOCOLLECT)` does not include collection. You can use `COPY/COLLECT` to override this. Copying a dump that already contains an appended collection always includes that collection.

For all file and unwind data to be collected successfully, all disks that were mounted at the time of the system crash should be remounted and accessible to the process running SDA. If SDA is invoked early on during startup to save the contents of the dump, for example, using `CLUE$SITE_PROC` (see Section 2.2.3), but disks are not mounted until a batch job is run, then the `COPY/NOCOLLECT` command should be used in the `CLUE$SITE_PROC` command procedure. Once all disks are mounted, a `COPY/COLLECT` can be used to save file and unwind data.

If the `COPY` and the `COLLECT` commands cannot be issued as a single step, a `COLLECT/SAVE` command writes the collection to a separate file that can be used later with the dump file. A later `COPY` command combines the two files.

4.29.10 DUMP Command

In Chapter 4, the following changes should be made to the `DUMP` command description:

- At the end of the format section, `/BYTE | /WORD |` should appear before `/LONGWORD` and `[/NOSUPPRESS]`.

- The following text should be appended to the description of the range parameter:

  The length of the range must be an exact multiple of the data item size (or of the index array size if /INDEX_ARRAY is specified).

- The following text should be appended to the description of the /RECORD_SIZE parameter:

  If no record size is given, and the length of the range is not more than 512 bytes, a single record is output containing the range specified, with no record number field. The length of the range must be an exact multiple of the data item size (or of the index array size if /INDEX_ARRAY is specified).

4.29.11 EVALUATE Command

In Chapter 4, the following changes should be made to the EVALUATE command description:

- In the Format section, [= filter] should be added after / [NO] SYMBOLS.
- The /SYMBOLS and /NOSYMBOLS qualifier descriptions should be replaced with the following:

  The default behavior of the EVALUATE command is to display up to five symbols that are known to be equal to the evaluated expression. If /SYMBOLS is specified with no filter, all symbols are listed in alphabetical order. If /NOSYMBOLS is specified, only the hexadecimal and decimal values are displayed. If /SYMBOLS is specified with a filter, only symbols that match the filter are displayed. The filter is a string containing wildcards, such as PCB$*.

4.29.12 MAP Command

In Chapter 4, the following text should be appended to the end of the description section:

On OpenVMS for Integrity servers, the MAP command can also provide additional data for addresses in system space. If the address is determined to be in a code section of an executive loaded image or a resident shareable image, and if the image file is accessible and was linked /TRACEBACK, then the traceback data is used to obtain and display module name and routine name information.

In Chapter 4, the following example should be added to the MAP command description:

```
11. SDA> EVALUATE 2F0/SYMBOL=PCB*
        Hex = 00000000.000002F0   Decimal = 752        PCB$L_INITIAL_KTB
                                                    PCB$L_PCB
```

This example shows the use of the symbol filter. Only symbols whose value is 2F0 and whose names begin with PCB are displayed.

4.29.13 SET CPU Command

In Chapter 4, the following lines should be appended to the list of SET PROCESS and SHOW PROCESS commands:

VALIDATE PROCESS/POOL *process-name*
VALIDATE PROCESS/POOL/ADDRESS=*pcb-address*
VALIDATE PROCESS/POOL/INDEX=*nn*
VALIDATE PROCESS/POOL/NEXT

4.29.14 SET PROCESS Command

In Chapter 4, the following lines should be appended to the list of SET PROCESS and SHOW PROCESS commands:

VALIDATE PROCESS/POOL *process-name*
VALIDATE PROCESS/POOL/ADDRESS=*pcb-address*
VALIDATE PROCESS/POOL/INDEX=*nn*
VALIDATE PROCESS/POOL/NEXT
VALIDATE PROCESS/POOL/SYSTEM

4.29.15 READ Command

In Chapter 4, the following text should be appended to the end of the description section:

In the description of the `directory-spec` parameter, `SYS$LOADABLE_IMAGES` and `SYS$LIBRARY` should be changed to `SYS$LOADABLE_IMAGES`, `SYS$LIBRARY`, and `SYS$SYSTEM`.

In Table 4-1, footnote 3, the following sentence should be removed:

These are found in `SYS$SYSTEM`, and are not automatically read in when you issue a `READ/EXEC` command.

4.29.16 SEARCH Command

The following changes should be made to the `SEARCH` command description in Chapter 4:

- The following sentence should be appended to the end of the `/LENGTH` and `/MASK` qualifier descriptions:

  This qualifier is ignored for string searches.

- In the `/STEPS` qualifier description, the text "or the given string" should be inserted after "expression."
- At the end of the final sentence of the `/STEPS` qualifier description, "...for value searches, and a step factor of a byte for string searches" should be added.
- In the Description section, the text "or string" should be inserted after "value."

4.29.17 SHOW_CALL_FRAME Command

In Chapter 4, the following text should replace the description of the `starting-address` parameter:

On Alpha:

An expression representing the starting address of the procedure call frame to be displayed. If no value for starting-address is given, the default starting address is the contents of the frame pointer (FP) register of the SDA current process. For a process that uses pthreads, the following SDA command can be used to display the starting addresses for all pthreads:

```
SDA> pthread thread -o u
```

On OpenVMS for Integrity servers:

An expression representing one of the following:

- The invocation context handle of a frame

- The address of an exception frame. This is equivalent to the SDA command:

  ```
  SDA> SHOW CALL_FRAME /EXCEPTION_FRAME=starting-address
  ```

- The address of a thread environment block (TEB). For a list of all TEBs for the process, use the following SDA command:

  ```
  SDA> pthread thread -o u
  ```

  If no `starting-address` is given, the default starting address is the invocation context handle of the current procedure in the SDA current process.

4.29.18 SHOW CLUSTER Command

In Chapter 4, the following changes should be made to the SHOW CLUSTER command description:

In the Format section, "/CIRCUIT=*pb-addr* |" should be placed before "/CSID."

In the qualifier description for /ADDRESS, "/CSID=*csid* and /NODE=*name*" should be replaced with "/CIRCUIT=*pb-addr*, /CSID=*csid*, and /NODE=*name*".

In the qualifier descriptions for /CSID and /NODE, "/CIRCUIT=*pb-addr*," should be added after "/ADDRESS=n".

The following text should be appended to the Description:

If the qualifier /CIRCUIT=*pb-addr* is specified, then the SHOW CLUSTER command displays only the information from the specified path block.

4.29.19 SHOW CRASH Command

The following changes should be made to the SHOW CRASH command description in Chapter 4:

The Format section should read, "SHOW CRASH [/ALL | /CPU = n]".

In the Description section the following should be appended to the second-to-last paragraph:

Unless /ALL is specified, the registers (on Alpha) or exception frame contents (on I64) are omitted from the display for any CPUs with CPUEXIT or DBGCPUEXIT bugchecks.

In the Description section, the phrase "and additionally displays all CPU database addresses in system dumps" should be removed from the final paragraph.

The command in example 3 should be changed to the following:

```
SDA> SHOW CRASH /ALL
```

4.29.20 SHOW DUMP Command

In Chapter 4, the following changes should be made to the SHOW DUMP command description:

The Format section should be replaced with the following:

```
SHOW DUMP [ /ALL
   | /BLOCK [ = m [ { : | ; } n ] ]
   | /COLLECTION [ = { ALL | n } ]
   | /COMPRESSION_MAP [ = m [ : n [ : p ] ] ]
   | /ERROR_LOGS
   | /FILE = { COLLECTION | DUMP }
   | /HEADER
   | /LMB [ = { ALL | n } ]
   | /MEMORY_MAP
   | /SUMMARY ]
```

In the Description section, the phrase "the memory map, and the file identification, and/or unwind data collection" should replace the phrase "and the memory map".

The following example should be appended to the SHOW DUMP command:

```
3. SDA> SHOW DUMP/COLLECTION

File and unwind data collection
------------------------------

Collection start VBN:     0002155B
Collection end VBN:       00022071
Collection block count:   00000B17

  VBN    Blocks>      Contents
-------- -------- ------------------------------
0002155B 000000C1 Unwind data segment 00000001 of _$30$DKB200:[VMS$COMMON.SYSEXE]DCL.EXE;1
0002161C 00000001 Unwind data segment 00000001 of _$30$DKB200:[VMS$COMMON.SYSEXE]USB$UCM_SERVER.EXE;1
0002161D 0000000C Unwind data segment 00000008 of _$30$DKB200:[VMS$COMMON.SYSEXE]USB$UCM_SERVER.EXE;1
.
.
.
0002200F 0000001F Unwind data segment 00000007 of _$30$DKB200:[VMS$COMMON.SYSEXE]LATACP.EXE;1
0002202E 00000006 Unwind data segment 0000000B of _$30$DKB200:[VMS$COMMON.SYSEXE]LATACP.EXE;1
00022034 00000001 Unwind data segment 00000002 of _$30$DKB200:[BISHOP]CMEXEC_LOOP.EXE;1
00022035 00000001 File data for _$30$DKA0:
00022036 0000003B File data for _$30$DKB200:
00022071 00000001 Disk data
```

This example of the SHOW DUMP/COLLECTION command shows the contents of the file identification and/or unwind data collection appended to a system dump when it was copied with the SDA command COPY/COLLECT. Note that unwind data segments are found only in system dumps from OpenVMS I64 systems.

4.29.21 SHOW PROCESS Command

In Chapter4, the following changes should be made to the SHOW PROCESS command description:

In the Format section:

- In "{ =P0 | =P1 | =ALL }", "=IMGACT" before "=ALL" should be inserted, and "(D)" should be added to "=ALL".
- In "[/BRIEF { /FREE | /UNUSED } |...]", "/CHECK" should be inserted after "/BRIEF".
- "/UNWIND_TABLE [=ALL]" should be changed to "/UNWIND_TABLE [= { ALL | name }]".

In the description of qualifier /ALL, "/STATISTICS" should be added to the line "/POOL/HEADER/RING_BUFFER".

The heading and the first paragraph in the description for the /POOL qualifier should read:

/POOL [= { P0 | P1 | IMGACT | ALL (D) }]

Displays the dynamic storage pool in the process´s P0 (process) region and/or the P1 (control) region and/or the image activator´s reserved pages, or optionally, a range of addresses. The default action is to display all dynamic storage pools.

The following should replace the description for the /UNWIND_TABLE qualifier:

/UNWIND_TABLE [= { ALL | name }] (I64 only)

If specified without a keyword, displays the master unwind table for the process. SHOW PROCESS/UNWIND=ALL displays the details of every process unwind descriptor. SHOW PROCESS/UNWIND=name displays the details of every unwind descriptor for the named image(s) (wildcards allowed). To look at unwind data for a specific PC in process space, use SHOW UNWIND address.

If some or all unwind data for an image was not included in the system dump (for example, it was not in the working set of the process at the time of the system crash), a SHOW PROCESS/UNWIND command may fail with a %SDA-W-NOREAD error because the unwind data is inaccessible. Collecting unwind data (see Commands COLLECT and COPY/COLLECT) will not correct this, as the collected unwind data is only used by SHOW UNWIND address and SHOW CALL.

The second bullet under "For I64" should read as follows:

Special purpose registers (PC, PSR, ISR). Note that the PC is the combination of the IP and the slot number from the PSR.

4.29.22 SHOW RESOURCES Command

The following changes should be made to the SHOW RESOURCES command description in Chapter 4:

The following keywords should be added to the /STATUS qualifier:

Keyword	Meaning
RM_FORCE	Forced tree move
RM_FREEZE	Freeze resource tree on this node
RM_INTEREST	Remaster due to master having no interest
XVAL_VALID	Last value block was long block

4.29.23 SHOW SPINLOCKS Command

The following changes should be made to the SHOW SPINLOCKS command description in Chapter 4:

The Format section should read as follows:

```
SHOW SPINLOCKS {[name]|/ADDRESS = expression|/INDEX = expression }
               [/OWNED|/DYNAMIC|/STATIC
               |/CACHED_PCB|/MAILBOX|/PCB|/PORT|/PSHARED]
               [{/BRIEF|/COUNTS|/FULL}]
```

In the description of the *name* parameter, the phrase "PCB, or cached PCB" should be changed to "PCB, cached PCB, or process-shared."

In the description of the /ADDRESS qualifier, the phrase "PCB, or cached PCB" should be changed to "PCB, cached PCB, or process-shared."

In the description of the /DYNAMIC qualifier, the phrase "PCB, or cached PCB" should be changed to "PCB, cached PCB, or process-shared." The following new qualifier should be added after the /PORT qualifier:

/PSHARED

Displays all process-shared (Pthreads) spinlocks.

In the third paragraph of the description section, the phrase "PCB and cached PCB" should be changed to "PCB, cached PCB, and process-shared."

4.29.24 SHOW MEMORY Command

In Chapter 4,the following changes should be made to the SHOW MEMORY command description:

In the /SLOTS qualifier description, the word "partition" should be "process."

4.29.25 SHOW GCT Command

The following changes should be made to the SHOW GCT command description in Chapter 4:

In the Format section, "[CHILDREN] |" should be "[/CHILDREN] |," and should be followed by "[/FULL] |".

In the Qualifiers section, the /FULL qualifier description should be added after /CHILDREN:

/FULL

When used with /CHILDREN, /OWNER=*n*, or /TYPE=*type*, the /FULL qualifier causes SDA to provide a detailed display of each node.

In the /OWNER and /TYPE descriptions, "Provides a detailed display of all nodes" should be replaced by "Displays all nodes".

In the /TYPE description, "CORE," "SOCKET," and "THREAD" should be added (in alphabetical order) to the list of valid types.

4.29.26 SHOW CPU Command

The following should be appended to the SHOW CPU command description in Chapter 4:

- On I64, the Exception Frame Summary

4.29.27 SHOW SYMBOL Command

The following text should be appended to the *symbol-name parameter* description:

Symbols that include lowercase letters must be enclosed in quotes.

4.29.28 SHOW UNWIND Command

The Format section should read as follows:

/IMAGE

SHOW UNWIND [address|/ALL|/IMAGE=name]

4.29.29 SHOW SWIS Command

The following changes should be made to the SHOW SWIS command description in Chapter 4:

- The Format section should be changed to the following:

```
SHOW SWIS [/RING_BUFFER [/CPU = (m, n,...)]]
```

- The description for the /CPU qualifier should be changed to:

```
/CPU = ( m, n,...)
```

  When used with /RING_BUFFER, displays only the entries for the specified CPU(s). If only one CPU is specified, the parentheses can be omitted.

- The final sentence of the Description section should be changed to:

  If you specify /CPU = (m, n), only the records for the specified CPU(s) are displayed.

4.29.30 CLUE CONFIG Command

The following qualifiers to the CLUE CONFIG command description in Chapter 5:

/CPU

Displays only the part of the system configuration that contains information about the system, memory and CPUs.

/ADAPTER

Displays only the part of the system configuration that contains information about the adapters and devices on the system.

The following sentence should be appended to the Description section:

If no qualifier is specified, the entire system configuration is displayed.

4.29.31 CLUE REGISTER Command

In Chapter 5, the following changes apply to the CLUE REGISTER command description:

The Format section should read as follows:

```
CLUE REGISTER [/CPU [cpu-id|ALL]
               | /PROCESS[/ADDRESS=n|INDEX=n
               | /IDENTIFICATION=n|process-name|ALL]]
```

The parameters and qualifiers are the same as those used for the CLUE_CALL_ FRAME command.

The following sentence should be appended to the Description section:

If neither /CPU nor /PROCESS is specified, the parameter (*CPU-id* or *process-name*) is ignored and the registers for the SDA current process are displayed.

4.29.32 I64 ISD_Labels Index Table

The following table replaces the I64 ISD_Labels Index Table in Chapter 10:

Table 4–5 I64 ISD_Labels Index

Index	Name	Meaning
0	SDA_CIO$K_FIX	Fixup
1	SDA_CIO$K_PROMO_CODE	Promote (code)
2	SDA_CIO$K_PROMO_DATA	Promote (data)
3	SDA_CIO$K_INIT_CODE	Initialization (code)
4	SDA_CIO$K_INIT_DATA	Initialization (data)
5	SDA_CIO$K_CODE	Code
6	SDA_CIO$K_SHORT_RW	Short data (read/write)
7	SDA_CIO$K_SHORT_RO	Short data (read only)
8	SDA_CIO$K_RW	Data (read/write)
9	SDA_CIO$K_RO	Data (read only)
10	SDA_CIO$K_SHORT_DZ	Short data (demand zero)
11	SDA_CIO$K_SHORT_TDZ	Short data (trailing demand zero)
12	SDA_CIO$K_DZERO	Demand zero
13	SDA_CIO$K_TR_DZERO	Trailing demand zero

4.29.33 Compiling and Linking an SDA Extension

In Section 10.2.1 Note 2, both instances of "ALPHA$LIBRARY" should be changed to "SYS$LIBRARY."

4.29.34 Debugging an Extension

In the example in Section 10.3, both instances of "alpha$library" should be changed to "sys$library."

4.29.35 Callable Routines Overview

In Section 10.4, "SDA$NEWPAGE" should be changed to "SDA$NEW_PAGE."

The following routines should be added alphabetically to the existing list:

SDA$CBB_BOOLEAN_OPER
SDA$CBB_CLEAR_BIT
SDA$CBB_COPY
SDA$CBB_FFC
SDA$CBB_FFS
SDA$CBB_INIT
SDA$CBB_SET_BIT
SDA$CBB_TEST_BIT
SDA$DELETE_PREFIX
SDA$FID_TO_NAME
SDA$GET_FLAGS

The final paragraph of the section that begins with "So, for example," should be part of the final bullet that begins with, "Some routines expect..."

The following new bullet should be added to the end of the section:

- The Common Bitmask Block (CBB) routines, SDA$CBB_*xxx*, are designed for use with local copies of the CBB structures that describe the CPUs in use in a system. The CBB structures are assumed to be at least CBB$K_STATIC_ BLOCK bytes in length. The definitions of the various CBB constants and field names used by these routines can be found in CBBDEF.H in SYS$LIBRARY:SYS$LIB_C.TLB.

  The set of routines is not intended to be an exhaustive set of all possible CBB-related operations, but provides those operations known to be needed. They might not work as expected with CBB structures set up for any purpose other than to describe CPUs.

4.29.36 Setting Up the Target System for Connections

In Section 11.3, in the Boot Command description, the phrase "with boot command" should be changed to "with the boot command."

In the paragraph before the SCD Configuration File description, the final sentence should read, "See the Boot Option Maintenance Menu, as described in the *HP OpenVMS System Manager's Manual, Volume 1: Essentials*.

In Section 11.3, the first bullet in the XDELTA Commands Description should read:

- n\xxxx\;R

The System Parameters description in Section 11.3 should have the following two bullets appended:

- **BREAKPOINTS**

  This parameter is a bitmask that enables existing INI$BRK calls within OpenVMS in the following situations:

  - Bit 0: At the start of INIT
  - Bit 1: At the end of INIT
  - Bit 2: At the point in INIT just prior to starting secondary CPUs
  - Bit 3: If INI$BRK is called from an outer mode
  - Bit 4: Before calling the initialization routine of a newly loaded executive image
  - Bits 5-31: Reserved by HP

  Note

  1. Calling INI$BRK from executive mode when bit 3 of BREAKPOINTS is not set results in process exit, or a SSRVEXCEPT bugcheck (if SYSTEM_CHECK or BUGCHECKFATAL is also set).
  2. Changing BREAKPOINTS from its default value of 3 may allow the security of the system to be compromised, and should only be used with caution.

- **TIME_CONTROL**

This parameter is a bitmask, disabling certain time control functions within OpenVMS:

- Bit 0: Disables system clock
- Bit 1: Disables CPU sanity timeouts
- Bit 2: Disables CPU spinwait timeouts

When XDELTA or SCD is loaded (bit 1 or bit 15 of boot flags is set), the value of TIME_CONTROL is changed from its default of zero to 6 (disable CPU sanity and CPU spinwait timeouts). This is to prevent these timeouts from occurring when the system is waiting at a breakpoint. If necessary, these settings can be altered using SYSGEN or the `DEPOSIT` command within XDELTA or SCD. Bit 0 should never be set.

At the end of Section 11.3.2, the following should be added:

The equivalent technique on I64 is as follows: Boot the system with only the SCD flag set (bit 15). When you see that the error has occurred, press Ctrl/P at the console. This action gives control to XDELTA (even though the XDELTA boot flag is not set), and you can now type 1;R. The target kernel will get control and wait for a connection for SCD.

Also, the following new Section 11.3.3 should be added:

The target kernel must have exclusive use of its ethernet device. Some system components, such as DECnet, will not start if the System Code Debugger is loaded. If there are multiple Ethernet devices, and the system is configured to give exclusive access of the SCD Ethernet device to the target kernel, the logical name DBGTK$OVERRIDE should be defined, indicating that the affected system components should start up as normal. The logical name can either be defined systemwide, or in the process where the startup command for the system component will be executed.

In Section 11.11.1, the final sentence of the second paragraph ("To remove symbols...") should be removed.

In Section 11.11.3, the second and third bullets should be removed, and the first bullet ("Access to All Executive Image Symbols") should not be a bullet.

In Section 11.12, the following new paragraph should appear immediately before Example 11-1:

Note that the example displays from Example 11-5 onwards are all taken from an OpenVMS I64 system. On an OpenVMS Alpha system, some of the output is different, but the commands entered are the same on both platforms, with one exception as noted in the accompanying text.

Also in Section 11.12, the references to "V8.2-014" should be changed to "V8.3-003."

Example 11-5 should be replaced with the following:

```
DBG> connect %node_name TSTSYS
%DEBUG-I-INIBRK, target system interrupted
DBG> show image
 image name                      set    base address         end address

 ERRORLOG                        no     0000000000000000     FFFFFFFFFFFFFFFF
 EXEC_INIT                       no     0000000000000000     FFFFFFFFFFFFFFFF
 SYS$ACPI                        no     0000000000000000     FFFFFFFFFFFFFFFF
*SYS$BASE_IMAGE                  yes    0000000000000000     FFFFFFFFFFFFFFFF
 SYS$DKBTDRIVER                  no     0000000000000000     FFFFFFFFFFFFFFFF
 SYS$DKBTDRIVER                  no     0000000000000000     FFFFFFFFFFFFFFFF
 SYS$DKBTDRIVER                  no     0000000000000000     FFFFFFFFFFFFFFFF
 SYS$EGBTDRIVER                  no     0000000000000000     FFFFFFFFFFFFFFFF
 SYS$OPDRIVER                    no     0000000000000000     FFFFFFFFFFFFFFFF
 SYS$PKMBTDRIVER                 no     0000000000000000     FFFFFFFFFFFFFFFF
 SYS$PKMBTDRIVER                 no     0000000000000000     FFFFFFFFFFFFFFFF
 SYS$PKMBTDRIVER                 no     0000000000000000     FFFFFFFFFFFFFFFF
 SYS$PLATFORM_SUPPORT            no     0000000000000000     FFFFFFFFFFFFFFFF
 SYS$PUBLIC_VECTORS              no     0000000000000000     FFFFFFFFFFFFFFFF
 SYS$SRBTDRIVER                  no     0000000000000000     FFFFFFFFFFFFFFFF
 SYSTEM_DEBUG                    no     0000000000000000     FFFFFFFFFFFFFFFF
 SYSTEM_PRIMITIVES               no     0000000000000000     FFFFFFFFFFFFFFFF
 SYSTEM_SYNCHRONIZATION          no     0000000000000000     FFFFFFFFFFFFFFFF

 total images: 18
DBG>
```

Example 11-7 should be replaced with the following:

```
DBG> set image system_debug
%DEBUG-I-DYNLNGSET, setting language IMACRO
DBG> show module
module name                     symbols   language   size

AUX_TARGET                      no        C             0
BUFSRV_TARGET                   no        C             0
BUGCHECK_CODES                  no        BLISS         0
C_TEST_ROUTINES                 no        C             0
LIB$$UNWIND_WEAK                no        BLISS         0
LIB$EF                          no        IMACRO        0
LIB$MALLOC                      no        C             0
LIB$MALLOC_64                   no        C             0
LINMGR_TARGET                   no        C             0
OBJMGR                          no        C             0
PLUMGR                          no        C             0
POOL                            no        C             0
PROTOMGR_TARGET                 no        C             0
SOCMGR                          no        C             0
SYS$DOINIT                      yes       IMACRO   122526
TMRMGR_TARGET                   no        C             0

total modules: 16
DBG> set module c_test_routines
DBG> show module c_test_routines
module name                     symbols    size

C_TEST_ROUTINES                 yes        5672
```

```
total C modules: 1
DBG> set language c
DBG> show symbol test_c_code*
routine C_TEST_ROUTINES\test_c_code
routine C_TEST_ROUTINES\test_c_code2
routine C_TEST_ROUTINES\test_c_code3
routine C_TEST_ROUTINES\test_c_code4
routine C_TEST_ROUTINES\test_c_code5
DBG> set break test_c_code
```

In Example 11-8, the final line should be replaced with the following:

```
113:     x = c_test_array[0];
```

Example 11-9 should be replaced with the following:

```
DBG> Set Mode Screen; Set Step Nosource

- SRC: module C_TEST_ROUTINES -scroll-source-----------------------------------
    98:     c_test_array[5] = in64;
    99:     c_test_array[6] = in32;
   100:     if (c_test_array[9] > 0)
   101:         *pVar =  (*pVar + c_test_array[17]) & c_test_array[9];
   102:     else
   103:         *pVar = (*pVar + c_test_array[17]);
   104:     c_test_array[7] = test_c_code3(10);
   105:     c_test_array[3] = test;
   106:     return c_test_array[23];
   107: }
   108: void test_c_code(void)
   109: {
   110:     int x,y;
   111:     __int64 x64,y64;
   112:
-> 113:     x = c_test_array[0];
   114:     y = c_test_array[1];
   115:     x64 = c_test_array[2];
   116:     y64 = c_test_array[3];
   117:     c_test_array[14] = test_c_code2(x64+y64,x+y,x64+x,&y64);
   118:     test_c_code4();
   119:     return;
   120: }
- OUT -output------------------------------------------------------------------




- PROMPT -error-program-prompt-------------------------------------------------


DBG>
```

In the paragraph immediately preceding Example 11-10, the reference to line 46 should be replaced with line 93.

Example 11-10 should be replaced with the following:

```
- SRC: module C_TEST_ROUTINES -scroll-source-----------------------------------
    80: void test_c_code4(void)
    81: {
    82:     int i,k;
    83:     for(k=0;k<1000;k++)
    84:       {
    85:          test_c_code5(&i);
    86:       }
    87:     return;
    88: }
    89: int test_c_code3(int subrtnCount)
    90: {
    91:     subrtnCount = subrtnCount - 1;
    92:     if (subrtnCount != 0)
    93:         subrtnCount = test_c_code3(subrtnCount);
    94:     return subrtnCount;
    95: }
    96: int test_c_code2(__int64 in64,int in32, __int64 test, __int64* pVar)
    97: {
    98:     c_test_array[5] = in64;
    99:     c_test_array[6] = in32;
   100:     if (c_test_array[9] > 0)
   101:         *pVar =  (*pVar + c_test_array[17]) & c_test_array[9];
   102:     else
- OUT -output------------------------------------------------------------------




- PROMPT -error-program-prompt-------------------------------------------------


DBG> Scroll/Up
DBG> set break %line 93
DBG>
```

Example 11-11 should be replaced with the following:

```
- SRC: module C_TEST_ROUTINES -scroll-source-----------------------------------
    82:     int i,k;
    83:     for(k=0;k<1000;k++)
    84:       {
    85:          test_c_code5(&i);
    86:       }
    87:     return;
    88: }
    89: int test_c_code3(int subrtnCount)
    90: {
    91:     subrtnCount = subrtnCount - 1;
    92:     if (subrtnCount != 0)
->  93:         subrtnCount = test_c_code3(subrtnCount);
    94:     return subrtnCount;
    95: }
    96: int test_c_code2(__int64 in64,int in32, __int64 test, __int64* pVar)
    97: {
    98:     c_test_array[5] = in64;
    99:     c_test_array[6] = in32;
   100:     if (c_test_array[9] > 0)
   101:         *pVar =  (*pVar + c_test_array[17]) & c_test_array[9];
   102:     else
   103:         *pVar = (*pVar + c_test_array[17]);
   104:     c_test_array[7] = test_c_code3(10);
- OUT -output-------------------------------------------------------------------
break at C_TEST_ROUTINES\test_c_code3\%LINE 93




- PROMPT -error-program-prompt-------------------------------------------------

DBG> Scroll/Up
DBG> set break %line 93
DBG> go
DBG>
```

In the paragraph immediately preceding Example 11-12, the reference to line 147 should be replaced with line 94, and the reference to line 146 with line 93.

Example 11-12 should be replaced with the following:

```
- SRC: module C_TEST_ROUTINES -scroll-source-----------------------------------
    82:     int i,k;
    83:     for(k=0;k<1000;k++)
    84:       {
    85:          test_c_code5(&i);
    86:       }
    87:     return;
    88: }
    89: int test_c_code3(int subrtnCount)
    90: {
    91:     subrtnCount = subrtnCount - 1;
    92:     if (subrtnCount != 0)
->  93:         subrtnCount = test_c_code3(subrtnCount);
    94:     return subrtnCount;
    95: }
    96: int test_c_code2(__int64 in64,int in32, __int64 test, __int64* pVar)
    97: {
    98:     c_test_array[5] = in64;
    99:     c_test_array[6] = in32;
   100:     if (c_test_array[9] > 0)
   101:         *pVar =  (*pVar + c_test_array[17]) & c_test_array[9];
   102:     else
   103:         *pVar = (*pVar + c_test_array[17]);
   104:     c_test_array[7] = test_c_code3(10);
- OUT -output-------------------------------------------------------------------
break at C_TEST_ROUTINES\test_c_code3\%LINE 93
break at C_TEST_ROUTINES\test_c_code3\%LINE 93
```

In the paragraph immediately preceding Example 11-13, the following statement should be added before the closing parenthesis:

The suffix _CODE0 is appended if the executive image is sliced.

Example 11-13 should be replaced with the following:

```
- SRC: module C_TEST_ROUTINES -scroll-source-----------------------------------
    82:     int i,k;
    83:     for(k=0;k<1000;k++)
    84:       {
    85:          test_c_code5(&i);
    86:       }
    87:     return;
    88: }
    89: int test_c_code3(int subrtnCount)
    90: {
    91:     subrtnCount = subrtnCount - 1;
    92:     if (subrtnCount != 0)
->  93:         subrtnCount = test_c_code3(subrtnCount);
    94:     return subrtnCount;
    95: }
    96: int test_c_code2(__int64 in64,int in32, __int64 test, __int64* pVar)
    97: {
    98:     c_test_array[5] = in64;
    99:     c_test_array[6] = in32;
   100:     if (c_test_array[9] > 0)
   101:         *pVar =  (*pVar + c_test_array[17]) & c_test_array[9];
   102:     else
   103:         *pVar = (*pVar + c_test_array[17]);
   104:     c_test_array[7] = test_c_code3(10);
- OUT -output-------------------------------------------------------------------
C_TEST_ROUTINES\test_c_code3\subrtnCount:       8
 module name     routine name     line           rel PC           abs PC
*C_TEST_ROUTINES test_c_code3       93       0000000000000DC0 FFFFFFFF800BAFC0
*C_TEST_ROUTINES test_c_code3       93       0000000000000DE0 FFFFFFFF800BAFE0
*C_TEST_ROUTINES test_c_code2      104       0000000000000F40 FFFFFFFF800BB140
*C_TEST_ROUTINES test_c_code       117       00000000000010B0 FFFFFFFF800BB2B0
                 XDT$INIT                    00000000000015C0 FFFFFFFF880955C0
*SYS$DOINIT      EXE$INITIALIZE   1973       0000000000000360 FFFFFFFF88094360
 SHARE$EXEC_INIT_CODE0                      000000000005C240 FFFFFFFF803BB640
 SHARE$EXEC_INIT_CODE0                      0000000000057F20 FFFFFFFF803B7320
 SHARE$EXEC_INIT_CODE0                      0000000000047850 FFFFFFFF803A6C50
 SHARE$EXEC_INIT_CODE0                      0000000000042E90 FFFFFFFF803A2290
- PROMPT -error-program-prompt-------------------------------------------------
DBG> set break %line 93
DBG> go
DBG> Step
DBG> examine subrtnCount
DBG> show calls
DBG>
```

In the paragraph immediately preceding Example 11-14, the reference to line 147 should be replaced with line 94, and the reference to line 146 with line 93.

Example 11-14 should be replaced with the following:

```
- SRC: module C_TEST_ROUTINES -scroll-source-----------------------------------
    83:     for(k=0;k<1000;k++)
    84:       {
    85:          test_c_code5(&i);
    86:       }
    87:     return;
    88: }
    89: int test_c_code3(int subrtnCount)
    90: {
    91:     subrtnCount = subrtnCount - 1;
    92:     if (subrtnCount != 0)
    93:         subrtnCount = test_c_code3(subrtnCount);
->  94:     return subrtnCount;
    95: }
    96: int test_c_code2(__int64 in64,int in32, __int64 test, __int64* pVar)
    97: {
    98:     c_test_array[5] = in64;
    99:     c_test_array[6] = in32;
   100:     if (c_test_array[9] > 0)
   101:         *pVar =  (*pVar + c_test_array[17]) & c_test_array[9];
   102:     else
   103:         *pVar = (*pVar + c_test_array[17]);
   104:     c_test_array[7] = test_c_code3(10);
   105:     c_test_array[3] = test;
- OUT -output-------------------------------------------------------------------
 module name     routine name     line           rel PC           abs PC
*C_TEST_ROUTINES test_c_code3       93       0000000000000DC0 FFFFFFFF800BAFC0
*C_TEST_ROUTINES test_c_code3       93       0000000000000DE0 FFFFFFFF800BAFE0
*C_TEST_ROUTINES test_c_code2      104       0000000000000F40 FFFFFFFF800BB140
*C_TEST_ROUTINES test_c_code       117       00000000000010B0 FFFFFFFF800BB2B0
                 XDT$INIT                    00000000000015C0 FFFFFFFF880955C0
*SYS$DOINIT      EXE$INITIALIZE   1973       0000000000000360 FFFFFFFF88094360
 SHARE$EXEC_INIT_CODE0                      000000000005C240 FFFFFFFF803BB640
 SHARE$EXEC_INIT_CODE0                      0000000000057F20 FFFFFFFF803B7320
 SHARE$EXEC_INIT_CODE0                      0000000000047850 FFFFFFFF803A6C50
 SHARE$EXEC_INIT_CODE0                      0000000000042E90 FFFFFFFF803A2290
stepped to C_TEST_ROUTINES\test_c_code3\%LINE 94
- PROMPT -error-program-prompt-------------------------------------------------
DBG> Step
DBG> examine subrtnCount
DBG> show calls
DBG> cancel break/all
DBG> go
DBG>
```

In the STEP/RETURN paragraph preceding Example 11-15, the phrase "on Alpha, or the R8 register on I64" should follow "R0 register".

Example 11-15 should be replaced with the following:

```
- SRC: module C_TEST_ROUTINES -scroll-source-----------------------------------
    83:     for(k=0;k<1000;k++)
    84:       {
    85:          test_c_code5(&i);
    86:       }
    87:     return;
    88: }
    89: int test_c_code3(int subrtnCount)
    90: {
    91:     subrtnCount = subrtnCount - 1;
    92:     if (subrtnCount != 0)
    93:         subrtnCount = test_c_code3(subrtnCount);
->  94:     return subrtnCount;
    95: }
    96: int test_c_code2(__int64 in64,int in32, __int64 test, __int64* pVar)
    97: {
    98:     c_test_array[5] = in64;
    99:     c_test_array[6] = in32;
   100:     if (c_test_array[9] > 0)
   101:         *pVar =  (*pVar + c_test_array[17]) & c_test_array[9];
   102:     else
   103:         *pVar = (*pVar + c_test_array[17]);
   104:     c_test_array[7] = test_c_code3(10);
   105:     c_test_array[3] = test;
- OUT -output-------------------------------------------------------------------
*C_TEST_ROUTINES test_c_code3       93       0000000000000DE0 FFFFFFFF800BAFE0
*C_TEST_ROUTINES test_c_code2      104       0000000000000F40 FFFFFFFF800BB140
*C_TEST_ROUTINES test_c_code       117       00000000000010B0 FFFFFFFF800BB2B0
                 XDT$INIT                    00000000000015C0 FFFFFFFF880955C0
*SYS$DOINIT      EXE$INITIALIZE   1973       0000000000000360 FFFFFFFF88094360
 SHARE$EXEC_INIT_CODE0                       000000000005C240 FFFFFFFF803BB640
 SHARE$EXEC_INIT_CODE0                       0000000000057F20 FFFFFFFF803B7320
 SHARE$EXEC_INIT_CODE0                       0000000000047850 FFFFFFFF803A6C50
 SHARE$EXEC_INIT_CODE0                       0000000000042E90 FFFFFFFF803A2290
stepped to C_TEST_ROUTINES\test_c_code3\%LINE 94
stepped on return from C_TEST_ROUTINES\test_c_code3\%LINE 94 to C_TEST_ROUTINES -
\test_c_code3\%LINE 94+17
C_TEST_ROUTINES\test_c_code3\%R8:       0
- PROMPT -error-program-prompt-------------------------------------------------
DBG> show calls
DBG> cancel break/all
DBG> go
DBG> step/return
DBG> examine r8
DBG>
```

In the first paragraph preceding Example 11-16, the phrase "for address 80002010" should be removed, and the phrase "this image or module" replaced with "INI$BRK."

The current Example 11-16 should be removed; the current Example 11-17 now becomes Example 11-16 and is replaced by the following:

```
- SRC: module C_TEST_ROUTINES -scroll-source-----------------------------------
    83:     for(k=0;k<1000;k++)
    84:       {
    85:          test_c_code5(&i);
    86:       }
    87:     return;
    88: }
    89: int test_c_code3(int subrtnCount)
    90: {
    91:     subrtnCount = subrtnCount - 1;
    92:     if (subrtnCount != 0)
    93:         subrtnCount = test_c_code3(subrtnCount);
->  94:     return subrtnCount;
    95: }
    96: int test_c_code2(__int64 in64,int in32, __int64 test, __int64* pVar)
    97: {
    98:     c_test_array[5] = in64;
    99:     c_test_array[6] = in32;
   100:     if (c_test_array[9] > 0)
   101:         *pVar =  (*pVar + c_test_array[17]) & c_test_array[9];
   102:     else
   103:         *pVar = (*pVar + c_test_array[17]);
   104:     c_test_array[7] = test_c_code3(10);
   105:     c_test_array[3] = test;
- OUT -output-------------------------------------------------------------------
 SYS$UTC_SERVICES               no     0000000000000000     FFFFFFFFFFFFFFFF
 SYS$VM                         no     0000000000000000     FFFFFFFFFFFFFFFF
 SYS$XFCACHE_MON                no     0000000000000000     FFFFFFFFFFFFFFFF
 SYSDEVICE                      no     0000000000000000     FFFFFFFFFFFFFFFF
 SYSGETSYI                      no     0000000000000000     FFFFFFFFFFFFFFFF
 SYSLDR_DYN                     no     0000000000000000     FFFFFFFFFFFFFFFF
 SYSLICENSE                     no     0000000000000000     FFFFFFFFFFFFFFFF
 SYSTEM_DEBUG                   yes    0000000000000000     FFFFFFFFFFFFFFFF
 SYSTEM_PRIMITIVES              no     0000000000000000     FFFFFFFFFFFFFFFF
 SYSTEM_SYNCHRONIZATION         no     0000000000000000     FFFFFFFFFFFFFFFF

 total images: 53
- PROMPT -error-program-prompt-------------------------------------------------
DBG> go
%DEBUG-I-INIBRK, target system interrupted
%DEBUG-I-DYNIMGSET, setting image SYS$BASE_IMAGE %DEBUG-W-SCRNOSRCLIN, No source line -
for address: FFFFFFFF80000310
DBG> show image
DBG> go
```

4.29.37 OpenVMS Alpha System Dump Debugger

The title to Chapter 12 should now read "OpenVMS System Dump Debugger," and the "Alpha Only" note should be removed.

The first paragraph in Section 12.4 should be replaced with the following:

If SDD cannot find one of the images through this search path, a warning message is displayed. SDD will continue initialization as long as it finds at least two images. If SDD cannot find the SYS$BASE_IMAGE and SYS$PUBLIC_ VECTORS files, which are the OpenVMS operating system's main image files, an error message is displayed and the debugger exits.

In Section 12.10, Steps 1 and 2 should be replaced with the following:

1. 1. Follow the steps in Section 11.12, up to and including Example 11-9 (Using The Set Mode Screen Command).
2. 2. Enter the following OpenVMS Debugger commands:

```
DBG> set break test_c_code5
DBG> go
DBG> deposit k=0
DBG> go
```

The following new paragraph should immediately precede Example 12-1:

Note that the example displays from Example 12-1 onwards are all taken from an OpenVMS I64 system. On an OpenVMS Alpha system, some of the output is different, but the commands entered are the same on both platforms.

In Example 12-1, the reference to "Alpha" should change to "I64," and "V7.2-019" to "V8.3-003".

Example 12-3 should be replaced with the following:

```
DBG> Set Mode Screen; Set Step Nosource

- SRC: module C_TEST_ROUTINES -scroll-source-----------------------------------
    67:
    68:     /* We want some global data cells */
    69: volatile __int64 c_test_array[34];
    70:
    71: void test_c_code5(int *k)
    72: {
    73:     int i;
    74:     char str[100];
    75:     for(i=0;i<100;i++)
    76:       str[i]= 'a';
    77:     str[99]=0;
->  78:     *k = 9;
    79: }
    80: void test_c_code4(void)
    81: {
    82:     int i,k;
    83:     for(k=0;k<1000;k++)
    84:       {
    85:          test_c_code5(&i);
    86:       }
    87:     return;
    88: }
    89: int test_c_code3(int subrtnCount)
- OUT -output-------------------------------------------------------------------




- PROMPT -error-program-prompt--------------------------------------------------


%DEBUG-I-SCRNOTORIGSRC, original version of source file not found for display in SRC
        file used is SYS$COMMON:[SYSHLP.EXAMPLES]C_TEST_ROUTINES.C;1
DBG>
```

Example 12-4 should be replaced with the following:

```
- SRC: module C_TEST_ROUTINES -scroll-source-----------------------------------
    67:
    68:     /* We want some global data cells */
    69: volatile __int64 c_test_array[34];
    70:
    71: void test_c_code5(int *k)
    72: {
    73:     int i;
    74:     char str[100];
    75:     for(i=0;i<100;i++)
    76:       str[i]= 'a';
    77:     str[99]=0;
->  78:     *k = 9;
    79: }
    80: void test_c_code4(void)
    81: {
    82:     int i,k;
    83:     for(k=0;k<1000;k++)
    84:       {
    85:          test_c_code5(&i);
    86:       }
    87:     return;
    88: }
    89: int test_c_code3(int subrtnCount)
- OUT -output-------------------------------------------------------------------
C_TEST_ROUTINES\test_c_code5\i: 100
C_TEST_ROUTINES\test_c_code5\k: 0
 module name     routine name     line           rel PC           abs PC
*C_TEST_ROUTINES test_c_code5       78       0000000000000CD0 FFFFFFFF800BAED0
*C_TEST_ROUTINES test_c_code4       85       0000000000000D60 FFFFFFFF800BAF60
*C_TEST_ROUTINES test_c_code       118       00000000000010D0 FFFFFFFF800BB2D0
                 XDT$INIT                    00000000000015C0 FFFFFFFF880955C0
*SYS$DOINIT      EXE$INITIALIZE   1973       0000000000000360 FFFFFFFF88094360
 SHARE$EXEC_INIT_CODE0                       000000000005C240 FFFFFFFF803BB640
 SHARE$EXEC_INIT_CODE0                       0000000000057F20 FFFFFFFF803B7320
 SHARE$EXEC_INIT_CODE0                       0000000000047850 FFFFFFFF803A6C50
 SHARE$EXEC_INIT_CODE0                       0000000000042E90 FFFFFFFF803A2290
- PROMPT -error-program-prompt--------------------------------------------------

%DEBUG-I-SCRNOTORIGSRC, original version of source file not found for display in SRC
        file used is SYS$COMMON:[SYSHLP.EXAMPLES]C_TEST_ROUTINES.C;1
DBG> examine i,k
DBG> show calls
DBG>
```

4.29.38 OpenVMS Alpha Watchpoint Utility

On the Part III page, the content in the "Alpha Only" note should be replaced with the following:

This utility runs on OpenVMS Alpha systems only.

On the first page of Chapter 13, the content in the "Alpha Only" note should be replaced with the following:

This utility runs on OpenVMS Alpha systems only.

4.29.39 SET PROCESS/LOG Command

In the format section, ARGS should be replaced with ARGUMENTS.

In description of parameter FLAGS, ARGS should be replaced with ARGUMENTS.

Note that these changes also affect the *HP OpenVMS DCL Dictionary: N–Z*.

5
Programming Release Notes

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

5.1 System Service Changes

V8.3

The following sections describe system service changes in this release.

5.1.1 Additions

Additions and changes have been made to the following system services:

- $GETJPI–item codes JPI$_TERMINAL and JPI$_SCHED_CLASS_NAME
- $GETRMI–description changed to: $GETRMI is an asynchronous system service and requires the $SYNCH service or another wait-state synchronous mechanism to guarantee that the required information is available. There is no synchronous wait form for this system service.
- $GETRMI item code change: RMI$_MODES returns the amount of time, in 10-ms units, spent by all currently active CPUs in all processor modes since the system was booted. Each increment in the returned time for a mode represents an additional 10 ms spent by the CPU in that mode. An active CPU is one that is actively participating in the processor scheduling that the OpenVMS instance performs. For additional information, see the $GETRMI system service in the *HP OpenVMS System Services Reference Manual*.
- $GETSYI–processor codes added for Intel Itanium© 2 and Intel© Itanium 3
- $GETSYI–item code definitions changed:

  SYI$_ACTIVE_CPU_MASK
  SYI$_CPUCONF
  SYI$_IO_PREFER_CPU
  SYI$_POTENTIAL_CPU_MASK
  SYI$_POWERED_CPU_MASK
  SYI$_PRESENT_CPU_MASK

- $INIT_VOL: some item codes now apply to ODS-5 as well as ODS-2

.

5.2 Range of Job-Limit Item Codes Increased for $GETQUI and $SNDJBC System Services

V8.3

In response to customer requests and to allow simpler definition of batch environments, the job limit item codes for two system services are increased in OpenVMS Version 8.3:

- For the $GETQUI system service, the range of the QUI$_JOB_LIMIT item code is now 1 to 65535.
- For the $SNDJBC system service, the range of the SJC$_JOB_LIMIT item code is now 1 to 65535.

5.3 Privileged Programs Might 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.

5.4 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.

5.4.1 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.

5.4.2 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.

5.4.3 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.

5.4.4 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.

5.4.5 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.

5.4.6 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.

5.4.7 Per-Thread Security Impacts Privileged Code and Device Drivers

Permanent Change

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.

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).

5.5 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`

5.5.1 IEEE Floating-Point Filter (I64 Only)

V8.3

In order to allow floating point exceptions to conform completely with IEEE-Std 754-1985, Intel provides a function called an IEEE filter. An application developer who wants to use this function can place a call to this function code within a normal OpenVMS exception handler. When an exception occurs, the filter can decode the floating point instructions that caused the exception, as well as decoding the IEEE rounding modes and precision, and determining the operands that caused the exception

To obtain a copy of this filter, access the following Intel Web site and look for the OpenVMS header:

http://www.intel.com/cd/software/products/asmo-na/eng/219748.htm

The application note available at this site explains the filter in more detail. The example source code and the filter object library are supplied as an OpenVMS backup save set.

Note that this filter is required only to make certain details of floating point exceptions conform to the IEEE standard. It is not required for normal floating point operation.

5.5.2 Limitation When Using Ctrl/C and STOP Button (OpenVMS Alpha)

V8.3

Condition: Pressing Ctrl/C to interrupt a program running under debugger control works only once. Thereafter, the Crtl/C interrupt is ignored. The same is true when using the DECwindows STOP button; the action is acknowledged only the first time the button is pressed.

Workaround: None.

5.5.3 Ada Event Support (I64 Only)

V8.3

Ada event support (SET BREAK/EVENT=*ada_event*, where *ada_event* is one of the events described by SHOW EVENT) is enabled on OpenVMS I64. However, this support is incomplete.

If you encounter problems with event breakpoints, switch to pthread events (SET EVENT_FACILITY *pthread*) as a workaround. Note that not all Ada events have an equivalent in the pthreads facility.

5.5.4 SHOW SYMBOL/TYPE Now Reports Correct Array Size (Alpha and I64)

V8.3

The SHOW SYMBOL command now reports the correct overall size of an array type. The previous behavior of SHOW SYMBOL/TYPE erroneously reported an incorrect size, for example:

```
DBG> sho sym/type z
data MATINV\Z
noncontiguous array descriptor type, 2 dimensions, bounds: [0:10,0:10], size: 4 bytes
cell type: atomic type, IEEE single precision floating point, size: 4 bytes
DBG>
```

The corrected behavior is:

```
DBG> sho symbol/type z
data MATINV\MATINV\Z
noncontiguous array descriptor type, 2 dimensions, bounds: [0:10,0:10], size: 484 bytes
cell type: atomic type, IEEE single precision floating point, size: 4 bytes
DBG>
```

5.5.5 EXAMINE/INSTRUCTION %PREVLOC Command Is Fixed (I64 Only)

V8.3

The EXAMINE/INSTRUCTION %PREVLOC command (and the EXAMINE/INSTRUCTION ^ alternative command) now works as expected. Previously, the debugger would not decrement the PC so that the same instruction was displayed.

5.5.6 SHOW MODULE Command Now Computes Module Size (I64 only)

V8.3

The SHOW MODULE command now reports a nonzero value for the size of a module. The value is an estimate, so you should use it only as a guide to compare the relative sizes of modules.

5.5.7 C++ Language Issues (I64 Only)

V8.2

Condition: The SHOW CALLS command sometimes displays C++ mangled names, rather than demangled names.

Workaround: None.

V8.3

Condition: The debugger does note support debugging C++ programs compiled with /OPTIMIZE.

Workaround: Compile C++ programs with /NOOPTIMIZE.

5.6 Ada Compiler(I64 Only)

V8.2

GNAT Pro (Ada 95) is available from AdaCore. Contact AdaCore at www.adacore.com or sales@adacore.com for more information.

5.7 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 *HP OpenVMS Utility Routines Manual* for more information about registering callback routines.

5.8 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.

5.9 C Run-Time Library

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

5.9.1 C RTL TCP/IP Header File Updates

V8.3

The C RTL ships header files for users to call TCP/IP. These headers have had numerous problems, making some of them unusable for anything beyond trivial TCP/IP programming.

Corrected headers shipped previously with several releases of TCP/IP and are located in their examples area. For OpenVMS Version 8.3, C RTL places those corrected headers into the C RTL header library (DECC$RTLDEF.TLB).

The following are examples of changes in the files:

- Member alignment is corrected. This returns the structures defined in the headers to the required packed alignment.
- Some constant macro definitions are changed to correctly match the constants with which TCP/IP is built.
- SOCKET.H changed MSG_DONTWAIT and MSG_DONTROUTE, swapping the values to be correct. To avoid hangs with the change, users might need to make changes if they coded workarounds for this.
- Previously, compatibility with older programs that were compiled with older headers was not possible because the older programs were not correctly passing data with TCP/IP. Data structure changes were made so users can communicate with TCP/IP. Without these changes, the header files were not usable. Users must recompile for these new definitions.
- The standards TCP/IP follows required header changes.
- New types, macros, and header files are added, as appropriate for new features in TCP/IP.

  C RTL header files changed:

```
BITYPES.H    IN.H        NAMESER_COMPAT.H   RESOLV.H
IF.H         IN6.H       NETDB.H            SOCKET.H
IF_ARP.H     INET.H      PCAP-BPF.H         STROPTS.H
IF_TYPES.H   NAMESER.H   PCAP.H             TCP.H
```

5.9.2 toascii Function Added

V8.3

The `toascii` function required by the X/Open specification had only a macro definition in `<ctypes.h>`.

This fix adds the `toascii` function, with an entry point in the DECC$SHR image.

5.9.3 64-Bit sigaction Problem Fixed

V8.3

Compile-time warnings resulted from using 64-bit pointers to the `sigaction` function.

This fix allows regular, long pointer (/POINTER=LONG), and short pointer (/POINT=SHORT) compilations.

5.9.4 64-Bit Pointer Capability Added to Several Math Functions

V8.3

The following C RTL math functions can now pass 64-bit pointers when a user compiles with the /POINTER_SIZE qualifier:

```
frexp       modf
frexpf      modff
frexpl      modfl
```

5.9.5 2-GB malloc No Longer Fails Silently

V8.3

The C RTL `malloc` function accepts an `unsigned int (size_t)` as its parameter. The `lib$vm_malloc` action accepts a (positive) signed integer as its parameter.

Allocating 2 GB (0x80000000) did not allocate the proper amount of memory and did not retun an error indication. A check is now added to the `malloc`, `calloc`, and `realloc` functions for sizes equal to or greater than 2 GB that fail the call.

5.9.6 Memory Leak in exec* Fixed

V8.3

A memory leak in `exec*` is fixed.

5.9.7 Behavior of exit after a Failed execl Fixed

V8.3

The C RTL contains a fix for the problem in which a call to `_exit` after a failed `execl` really exits but should not.

In the OpenVMS implementation of `vfork`, a child process is not actually started as it is started on most UNIX systems. However, the C RTL creates some internal data structures intended to mimic child-process functionality (called the "child context").

A bug occurred whereby after a `vfork` while in the child context, a call to an `exec` function justifiably fails, then calls `_exit`. On UNIX systems, after the failed `exec` call, the child process continues to execute. A subsequent call to `_exit` terminates the child. In the OpenVMS implementation, after the failed `exec` call, the child context terminates. A subsequent call to `_exit` terminates the parent.

The C RTL fix is enabled by a feature logical switch, DECC$EXIT_AFTER_ FAILED_EXEC. Enabling this feature logical allows the child context to continue execution.

With DECC$EXIT_AFTER_FAILED_EXEC disabled or not defined, the current behavior remains the default.

5.9.8 confstr Enhancements

V8.3

To conform to the X/Open specification, the `confstr` function now allows a zero-length buffer to be passed in.

Also, `confstr` now supports the following three HP-UX symbolic constants, which are added to header file `<unistd.h>`:

_CS_MACHINE_IDENT
_CS_PARTITION_IDENT
_CS_MACHINE_SERIAL

5.9.9 fopen Failure Fixed

V8.3

Previously, if a user called `fopen`(*file*, "wb+"), where *file* was a quoted DECNET specification and the remote file was absent, RMS reported a syntax error in the file specification instead of opening the file for update (creating a new version).

This problem is fixed in Version 8.3.

5.9.10 Possible File-Pointer-Locking Hang Condition

V8.3

OpenVMS Version 8.2 introduced file-pointer locking for multithreaded C programs (`flockfile` and friends). The C RTL internally locks file pointers to conform to the X/Open specification.

A problem occurred in instances where file pointers were not unlocked on an error return during the preloading of data.

Users who experience this problem might see application hangs in multithreaded programs using file-pointer I/O. Analysis of the program state shows a deadlock of threads on TIS-recursive mutexes used to implement the file-pointer locks.

This problem is fixed in Version 8.3.

5.9.11 Backport Library No Longer Shipped

V8.3

Previously included with the compiler distribution kit was a C RTL backport object library, which allowed developers on older versions of OpenVMS to use the latest C RTL functions. This backport object library is no longer being shipped.

5.9.12 Header File <time.h> Changes

V8.3

The following problem is fixed. Users who still experience this problem might have to recompile their application to see the corrected behavior.

The C RTL `<time.h>` header file defines the `struct tm` structure and the functions `gmtime`, `localtime`, `gmtime_r`, and `localtime_r`.

When the calling of one these functions and the application and the C RTL disagree about the size of `struct tm`, a user application can see data corruption or an access violation.

The tm structure has optional members for BSD extensions: tm_gmtoff and tm_zone. These are not defined in the ANSI or POSIX specifications or, for compatibility, with older compilations.

The previously mentioned functions have three different definitions in the C RTL:

- Local time (no prefixes)
- UTC time (after OpenVMS V7.0)
  - __UTC_ prefixes for no BSD extensions
  - __UTCTZ_ prefixes when using the BSD extensions

The __UTCTZ_ prefixed functions expect to assign only the longer tm structure with the BSD extensions defined.

The problem occurs when the <time.h> header file and the C RTL do not agree on the number of structure members in struc tm. For example, the problem occurs with a C++ compilation using a compile-time macro implying _ANSI_C_SOURCE (such as _POSIX_C_SOURCE), which maps the listed C RTL functions using __UTC_ prefixes. The functions expect the shorter tm data structure, but the user program uses the longer tm structure definition. The copy-back of data from the function return tries to access data not allocated by the C RTL for the tm data members. This can result in unpredictable behavior, such as an unintended memory or access violation.

In OpenVMS Version 8.3, changes are made to <time.h> to make sure the C RTL selects the appropriate prefixing for the listed functions.

5.9.13 Header File <time.h> Makes *_r Non-ANSI Functions Visible

V8.3

The C RTL functions ctime_r, gmtime_r, and localtime_r defined in the X/Open specification are not in the ISO/ANSI C standard and should not be visible when compiling only for ANSI compliance. Previously in the C RTL, they were visible.

This situation is fixed. Checks are added in the <time.h> header to make these functions visible only when not compiling for ANSI compliance.

5.9.14 Header File <decc$types.h>: time_t int Declaration

V8.3

The time_t structure defined in the <decc$types.h> header file has been historically declared as unsigned long int on OpenVMS systems. UNIX platforms tend to declare it as a signed type, which can cause problems for ported programs on OpenVMS.

For UNIX compatibility, this fix uses a compile macro __SIGNED_INT_TIME_T to declare time_t as int.

The default remains unsigned long int for compatibility with older programs.

5.9.15 New DECC$SHRP.EXE Image

V8.3

OpenVMS installs a new shareable image DECC$SHRP.EXE to implement C RTL functions requiring protected mode. This shareable image is installed on all Alpha and Integrity processors and is invoked from either the DECC$SHR.EXE or DECC$SHR_EV56.EXE shareable image.

5.9.16 Header File <wchar.h> and C++ %CXX-W-ENVIRSTKDIRTY Message

V8.3

The <wchar.h> header file is fixed to avoid the problem where the C++ Version 7.1 compiler can issue a %CXX-W-ENVIRSTKDIRTY warning when compiling with the _XOPEN_SOURCE macro defined.

5.9.17 Header File <builtins.h> __CMP_SWAP* and _Interlocked* Visible to C++

V8.3

The compare and swap builtins (__CMP_SWAP* and _Interlocked*) in <builtins.h> did not include the OpenVMS Alpha C++ compiler. Because HP C++ Version 7.1 requires them, a change in conditional compilation now makes these builtins visible.

5.9.18 Extra Parameters to fcntl Ignored

V8.3

Previously, calls to `fcntl` with a third-optional parameter when only two were expected (for commands not expecting a third parameter) returned errors.

This problem is fixed. Any unneeded third parameter is now ignored.

5.9.19 Problem with fwrite to stdout Fails with Large System MAXBUF

V8.3

Previously, using `fwrite` to `stdout` resulted in an error if the system's MAXBUF SYSGEN parameter was greater than or equal to 33278.

The following message might have been returned when using `perror` (as viewed by `perror`) after the failing `fwrite` call:

```
Error writing output: : non-translatable vms error code: 0x186A4

 %rms-f-rsz, invalid record size
```

This problem is fixed.

5.9.20 Problem with Read/Write for Socket Transfers Greater Than 64K

V8.3

Support is added for socket transfers greater than 64K bytes for the following socket routines:

```
send        recv         read
sendto      recvfrom     write
sendmsg     recvmsg
```

5.9.21 Problem with Nanosleep on I64 Systems

V8.3

On OpenVMS I64 systems, the `nanosleep` function used to return an invalid status of -1 and set `errno` to EINTR (interrupted system call) when the function was correctly sleeping.

This problem is fixed.

5.9.22 Builtin __fci Added for I64 Systems

V8.3

The <builtins.h> header file is updated with the prototype for the new __fci builtin (a builtin for the fc.i instruction) now supported by the HP C compiler.

5.9.23 _FAST_TOUPPER Macro Added

V8.3

As of OpenVMS Version 8.3, to comply with the C99 ANSI standard and X/Open Specification, the _tolower and _toupper macros by default do not evaluate their parameter more than once. They simply call their respective tolower or toupper function. This avoids side effects (such as i++ or function calls) where the user can tell how many times an expression is evaluated.

To retain the older, optimized behavior of the _tolower and _toupper macros, compile with /DEFINE=_FAST_TOUPPER. Then, as in previous releases, these macros optimize the call to avoid the overhead of a run-time call. However, the macro's parameter is evaluated more than once to determine how to calculate the result. this could create unwanted side effects.

5.9.24 Call to atof("NaN") no Longer Gives Arithmetic Trap

V8.3

On OpenVMS Alpha Version 8.2 and higher, the following call to atof("NaN") gave an arithmetic trap:

```
d = atof("NaN");
```

```
%SYSTEM-F-HPARITH, high performance arithmetic trap, Imask=00000000,
Fmask=00000002, summary=02, PC=FFFFFFFF80BB23D4, PS=0000001B
-SYSTEM-F-FLTINV, floating invalid operation, PC=FFFFFFFF80BB23D4,
PS=0000001B
```

This is fixed in OpenVMS Version 8.3. The arithmetic trap is no longer generated.

5.9.25 No New Entries for DECC$*.OLB Object Libraries

V8.3

As of OpenVMS Alpha and I64 Version 8.2, no new entry points are being added to the DECC$*.OLB object libraries. This means new C RTL entry points do not get mapped through these libraries to prefixed entries. For example, the new OpenVMS Version 8.3 entry point crypt, compiled with /PREFIX=NONE, does not get mapped from crypt to decc$crypt.

5.10 Calling Standard and Rotating Registers (I64 Only)

V8.3

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. That is, 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.

Previously, 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 interpreted their register number parameters without applying an adjustment for the effects of the register rename base and rotating size registers. This error is corrected.

5.11 Common Data Security Architecture (CDSA) Considerations

The following notes pertain to CDSA.

5.11.1 Secure Delivery

V8.3

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.

Secure Delivery is fully functional with OpenVMS Version 8.3. For more information, refer to the *HP OpenVMS Version 8.3 New Features and Documentation Overview*.

5.11.2 Installation and Initialization Considerations

V7.3-2

Installation of CDSA is done automatically when you install the operating system.

- When a new version of CDSA is installed separately from an Operating System upgrade, 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 upgrade procedure when the system is rebooted. You also do not need to add the upgrade procedure 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 V2.2 is referenced by DEC AXPVMS OPENVMS V8.3
-PCSI-E-HRDRF1, the two products are tightly bound by this software dependency
```

5.12 Debugging Modes: Avoiding CPUSPINWAIT Bugchecks

Permanent Condition

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 5.29.

5.13.1 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 5.10 for additional details.

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

Permanent Correction

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)

Permanent Condition

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 Potential Must-Be-Zero RMS Error: Making Room for New File Options in the FAB

V8.3

There is very little remaining unassigned space in the RMS user file access block (FAB). To allow for future growth in file-processing options implemented through the FAB, the last unassigned bit in the file-processing options field in the FAB (FAB$L_FOP) has been defined as the FAB$V_EXTEND_FOP option. This option will be used in the future to request and validate assignments to two new FAB fields that span previously unused bytes in the FAB:

- FAB$W_FOPEXT: The FOPEXT word field is reserved for the definition of new file-processing options that might require implementation in the future.
- FAB$W_RESERVED_MBZ: The RESERVED_MBZ word field is being reserved for future use. Its usage is currently open for future definition.

In a future release, when one or more new file-processing options are implemented, the FAB$V_EXTEND_FOP bit will have to be set in the FAB$L_FOP field to take advantage of any new option in the FOPEXT field. However, when it is set, it will also indicate that any undefined bits in the FOPEXT field are clear and FAB$W_RESERVED_MBZ contains a zero. If these conditions are not met when this bit is set, a fatal error (RMS-F-FOPEXTMBZ) will be returned to the user for any RMS file-level service.

The addition of the EXTEND_FOP option is fully upwardly compatible except if a user sets this last bit in the FAB$L_FOP field by accident and either of these two new FAB fields happens to be nonzero. Previously, RMS ignored this last bit in the FAB$L_FOP field if it was set by accident.

If the RMS-F-FOPEXTMBZ error is returned, you must clear either the EXTEND_FOP option in the FAB$L_FOP field or the two new fields (FAB$W_FOPEXT) and FAB$W_RESERVED_MBZ.

5.17 HP COBOL Run-Time Library (RTL)

V8.3

For OpenVMS Alpha and OpenVMS I64 Version 8.3, the HP COBOL RTL (DEC$COBRTL) has been updated to V2.8-781.

5.18 HP Fortran for I64

V8.3

The HP Fortran for OpenVMS Integrity servers 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 or greater.

HP Fortran for OpenVMS Integrity servers 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 V8.0 or V8.1 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 V8.0 or V8.1 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

V8.3

The following notes pertain to the HP MACRO for OpenVMS I64 compiler:

- Prior to OpenVMS Version 8.3, the compiler generated the wrong code for the HALT instruction. On I64 platforms, the HALT instruction is implemented using the Itanium `break` instruction with a reserved literal value of `BREAK$C_SYS_HALT`. Because of a bug in the compiler's build environment, the Macro-32 compiler used the wrong literal value. This problem has been fixed in Version 8.3. Any code that uses the HALT instruction should be recompiled with the Version 8.3 compiler. For systems prior to Version 8.3, the correct behavior can be accomplished as follows:

```
$BREAKDEF
IA64_HALT #BREAK$C_SYS_HALT ; Issue break instruction with correct literal
HALT                        ; Use HALT builtin to inform compiler that this ends the flow of control
```

- The compiler might optimize away instructions prior to HALT, BPT, and EVAX_BUGCHK. The optimizer does not identify these special instructions as implicitly reading all registers by either writing a dump file or transferring control to a debug environment. The apparently unused instructions are removed by mistake. The compiler will be taught about the special behavior of these instructions. Though there is no syntax to force arbitrary instructions to be included in the final object file, the IA64_LD8_A builtin can be used as a workaround. See the following Macro-32 paragraphs for information about this builtin. In addition, specifying /NOOPTIMIZE preserves the apparently unused instructions at the expense of slower and larger code.
- The compiler might optimize away apparently unused memory loads. The compiler's optimizer can recognize whether or not the result of a memory load is used. If the result appears to be unused, the optimizer removes the memory load as well. However, some code might be using the memory load to fault a page into memory before raising IPL for example. In these cases, the removed instruction prevents the page from being faulted into memory, and the subsequent code at high IPL experiences a fatal page fault at high IPL exception. Although there is no syntax to force arbitrary instructions to be included in the final object file, the IA64_LD8_A builtin can be used as a workaround. The IA64_LD8_A builtin, new in Version 8.3, generates a special form of the Itanium "ld8" instruction that also places the fetched address into the ALAT (Advanced Load Address Table). The compiler identifies this special form of "ld8" as having a side effect and that it cannot be removed from the final object file even if the result appears to be unused. The insertion of the address into the ALAT does not cause any problems or require any additional changes. For unused memory loads that must remain in final object file, change them from:

  ```
  MOVL (Rn),Rn
  ```

  into

  ```
  IA64_LD8_A Rn,(Rn),#0
  ```

  HP will add new syntax to the compiler in a future release to designate instructions that must survive to the final object file.

  For systems prior to Version 8.3, there is no IA64_LD8_A builtin. The only workaround is to use /NOOPTIMIZE.

5.19.2 HP MACRO for OpenVMS Alpha Systems

V8.3

The compiler might optimize away apparently unused memory loads. The compiler's new optimizer can recognize whether or not the result of a memory load is used. If the result appears to be unused, the optimizer removes the memory load as well. However, some code may be using the memory load to fault a page into memory before raising IPL for example. In these cases, the removed instruction prevents the page from being faulted into memory, and the subsequent code at high IPL experiences a fatal page fault at high IPL exception. The only workaround is either to use /NOOPTIMIZE or revert to the prior Macro-32 compiler, which does not contain the optimization.

The new Macro-32 compiler is named SYS$SYSTEM:MACRO.EXE and is the default image activated by the DCL MACRO command. The older compiler can be found at SYS$SYSTEM:ALPHA_MACRO.EXE. For the DCL command MACRO to use the older compiler, define a MACRO logical name as follows:

```
$ DEFINE MACRO SYS$SYSTEM:ALPHA_MACRO.EXE
```

5.19.3 /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.

5.19.4 /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.

5.19.5 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.

5.20 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 5.35.

5.20.1 Reporting a Problem to HP

Permanent Condition

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

Permanent Restriction

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

Permanent Restriction

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

Permanent Restriction

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

Permanent Restriction

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

5.20.8 Optimal Working Set Extent and Page File Quota Settings

Permanent Restriction

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)

Permanent Restriction

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. This is a permanent condition.

5.22.3 Librarian Fails to Report Errors When Process Quota Too Low

Permanent Restriction

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.

5.23 Analyze Utility for OpenVMS (I64 Only)

The following release notes pertain to the Analyze utility on OpenVMS I64.

5.23.1 Releasing Process Resources Fixed in I64 Analyze Image /SELECT

V8.3

For I64 image files, ANALYZE/SELECT did not release process resources; a wild card operation could fail with SECTBLFUL (or other exceeded quota messages).

This problem is fixed in Version 8.3.

5.23.2 Selective Output for Debug Line Information Fixed

V8.3

For I64 image files, ANALYZE/IMAGE/SELECT=DEBUG=LINE did not format the corresponding debug line information. The line selection only worked with /TRACE.

This problem is fixed in Version 8.3.

5.24 Command Definition Utility (I64 Only)

The following release notes pertain to the Command Definition utility on OpenVMS I64.

5.24.1 Record Attributes Fixed for I64 Images

V8.3

For I64 images (command tables), the Command Definition Utility (CDU) set record attributes, which caused warnings when copying or appending image files. Despite these warning messages, the copied or appended files contained correct content.

In Version 8.3, CDU sets compatible attributes that allow these file operations to complete without a warning message.

5.25 Linker Utility for OpenVMS Alpha

The following release notes pertain to the Alpha Linker Utility on all OpenVMS platforms. For I64 Linker release notes, see Section 5.26.

5.25.1 Linker Appears to Hang When Many Files Are Specified

V8.3

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$FIND_FILE may take a long time to complete and the linker may appear to hang. Depending on the number of files being linked and the use of logical names in their specification, the linker may take hours to finish because LIB$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 by pressing Ctrl/Y after the linker has called LIB$FIND_FILE.

To determine which file is missing, follow the steps described with the RMS_RELATED_CONTEXT= option in the *Linker Manual*, Part IV, LINK Command Reference.

5.25.2 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.

For more information, see the /LIBRARY qualifier in the *Linker Manual* Part IV, LINK Command Reference.

Shareable image libraries do not contain a copy of an image. They contain the image's name, the image's identification information, and a table including the image's universal symbols. The identification information is provided by the `GSMATCH=` option when the shareable image is linked. For more information, see the `GSMATCH=` option in the Linker Manual Part IV, LINK Command Reference.

It may happen that a shareable image is relinked but that a library is not updated. To handle such a case, the linker checks for compatibility. The linker makes the same check that the image activator does: that is, it uses the `GSMATCH` criteria to verify compatibility.

For VAX, the linker also compares against the date and time, signaling LINK-I-DATMISMCH, if they are different.

For Alpha, the initial behavior of linker was the same as the VAX linker. The check was seen as too sensitive and the default behavior was changed to use only the GSMATCH criteria. The old VAX behavior can be obtained by setting the logical name LINK$SHR_DATE_CHECK.

5.25.3 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.

5.26 Linker Utility for OpenVMS I64

V8.3

The following release notes pertain to the I64 Linker utility.

For Alpha Linker release notes, see Section 5.25.

5.26.1 Linker Writes Incorrect Interimage Debug Fixups into Debug Symbol File

V8.3

In some situations, the linker creates interimage fixups for the OpenVMS debugger. The interimage debug fixup is a result of compiler-generated debug relocations, which the linker cannot resolve. That is, the debugger requires these fixups to determine a value from a shareable image at run-time. Compilers might differ on how often they require the linker to create inter-image fixups for the debugger. The C compiler rarely uses inter-image debug fixups, but the C++ compiler often uses them. When linking such images with /DEBUG, the linker writes the debug information followed by the interimage debug fixups. With /NODSF (the default) the information is written correctly into the image file, but with /DSF the information is sometimes written incorrectly into the DSF file.

For example, the DEBUG informationals and the DEBUG error in the following sample occur because the linker has written the DSF file incorrectly:

```
$ RUN/DEBUG MINIREF
%DEBUG-I-INFODWARF, error reading Dwarf info: Section 0a extends outside file
%DEBUG-I-INFODWARF, error reading Dwarf info: Section 0c extends outside file
%DEBUG-I-INFODWARF, error reading Dwarf info: SHT_VMS_FIXUP section 10 size 17eb
e0 not multiple of 18
%DEBUG-I-INFODWARF, error reading Dwarf info: SHT_VMS_FIXUP section 12 size 17ec
30 not multiple of 18

         OpenVMS I64 Debug64 Version V8.3-003

%DEBUG-F-ACCVIO, access violation, reason mask=00, virtual address=000000000014A
000, PC=000000007BD73100, PS=0000001B
%DEBUG-I-INITIAL, Language: C, Module: MINIREF

DBG>
```

The image file is not affected; it can be executed with the RUN command without any problem:

```
$ RUN MINIREF
```

This error will be corrected in the next release of the I64 Linker:

5.26.2 OpenVMS I64 Object Module and Image File Information Currently Unavailable

V8.3

Information about the OpenVMS I64 extension to ELF (Executable and Linkable Format; the object module and image file format on I64) is not available at this time. It will be provided in a future release.

For information about the Alpha or VAX object language format, see the respective appendixes in the *OpenVMS Alpha/VAX Version 7.3 OpenVMS Linker Utility Manual*, available from the documentation bookshelf at the following URL:

```
http://h71000.www7.hp.com/doc/os732_index.html
```

5.26.3 /SELECTIVE_SEARCH Might Incorrectly Ignore Transfer Address

V8.3

If you have an I64 object module containing a transfer address and you include the module in the link operation with the /SELECTIVE_SEARCH qualifier, the linker will not find its transfer address.

In the following example, the object module (MAIN.OBJ) contains a transfer address but /SELECTIVE_SEARCH ignores it:

```
$ LINK MAIN/SELECTIVE_SEARCH
%ILINK-W-USRTFR, image USER:[JOE]MAIN.EXE;1 has no user transfer address
```

This condition happens only when I64 object modules, meant to provide the program's transfer address, are included in the link operation with the SELECTIVE_SEARCH attribute. The SELECTIVE_SEARCH attribute is given to an object module when you specify /SELECTIVE_SEARCH qualifier to the object module in the LINK command. For example,

```
$ LINK MAIN/SELECTIVE_SEARCH
```

A LIBRARY command,

```
$ LIBRARY/INSERT LIB.OLB MAIN.OBJ/SELECTIVE_SEARCH
```

when that module from the library is used to resolve references in a link operation, either implicitly,

```
$ LINK/EXECUTABLE=MAIN SUBROUTINES.OBJ, LIB.OLB/LIBRARY
```

or explicitly:

```
$ LINK/EXECUTABLE=MAIN SUBROUTINES.OBJ, LIB.OLB/INCLUDE=MAIN
```

This problem manifests itself in one of two ways:

- The linker displays where a warning message as in the preceding example. This condition occurs if no other object module in the link operation provides a transfer address, weak or otherwise.
- The linker does not display a message. Such a condition occurs if other object modules in the link operation provide a transfer address, weak or otherwise. If the linker fails to properly identify the transfer address from the selectively searched object module, it selects one from the other object modules. That transfer address unintentionally becomes the main entry point for the image, although this was not intended. Even though the map file does indicate the incorrect transfer module and transfer address the linker has assigned, the problem might not become evident until you actually run your application.

5.26.4 Differences Between the I64 Linker and the Alpha Linker

V8.3

Be sure to read the *HP OpenVMS Version 8.3 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 8.3 New Features and Documentation Overview* also contains a description of features that are new with the OpenVMS I64 Linker.

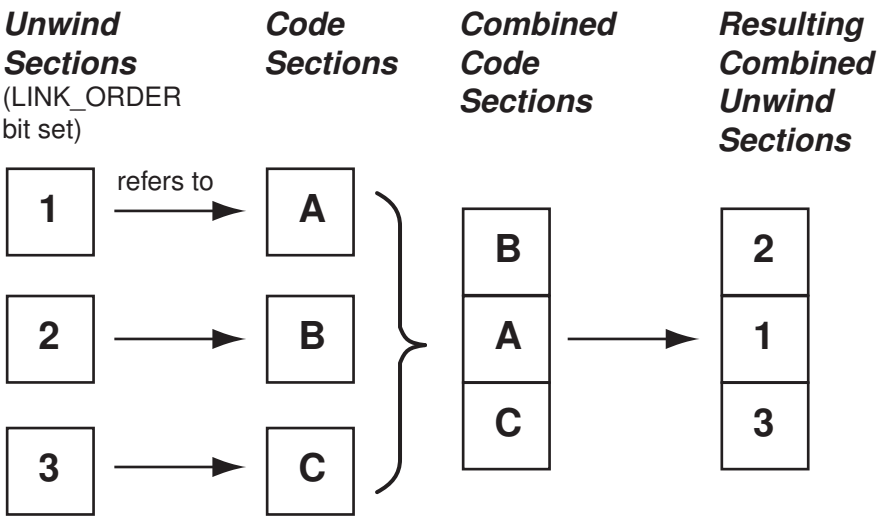
5.26.5 LINK_ORDER Section Header Flag Not Supported

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.



5.26.6 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. This is a permanent condition.

5.26.7 Error in Handling Initialized Overlaid Program Sections Fixed

V8.3

In previous versions of the I64 linker, initialized overlaid program sections with zeros are incorrectly seen as compatible initializations. In OpenVMS Version 8.2 and Version 8.2-1, the I64 Linker accepts sections initialized with zeros to be compatible with sections containing nonzero initial values. Under this condition, the order of modules in a link operation can result in different initial values in the image.

This problem is fixed in Version 8.3.

5.26.8 Removal of Linker Qualifiers /EXPORT_SYMBOL_VECTOR and /PUBLISH_GLOBAL_SYMBOLS

V8.3

Creating a shareable image by simply exporting all global symbols does not work. Previous I64 Linker versions provide the /EXPORT_SYMBOL_VECTOR and /PUBLISH_ GLOBAL_ SYMBOLS qualifiers to assist in creating symbol vector options for all global symbols on a per-module basis. However, using these qualifiers exports the same universal symbol from different images. Because you cannot exclude exporting the same C++ template from different images, this creates problems.

Both qualifiers are removed from the Version 8.3.

This problem will be fixed for the I64 Linker in a future release of OpenVMS for I64.

5.26.9 Support for Longer Symbol Names in Options

V8.3

For options, the linker internal line buffer is increased to hold 2048 characters. This allows specifying options with symbol names of the maximum supported length of 1024 characters.

5.26.10 Better Use of Memory for Linker-Created Code Stubs

V8.3

The I64 linker generates code stubs, which are visible when stepping through the code with the OpenVMS Debugger. Code can be inserted for calls and can have different sizes. However, the earlier linker added the code as fixed-size code sections using the maximum necessary code size.

The Version 8.3 linker writes the code stubs using the actual size. This eliminates unused space between the stubs and might also release unused memory.

5.26.11 Compiler Support for Demangled Symbol Names

V8.3

To make symbol names unique, name mangling is used for some programming languages. Starting with Version 8.3, the linker tries to display the demangled names, also known as source code names. These names are used in linker messages and, on request, in a translation table for mangled global symbols that appears in the linker map (see *HP OpenVMS Version 8.3 New Features and Documentation Overview*). This feature depends on compiler support for name demangling.

5.26.12 Maximum Number of Sections

V8.3

For more than 65280 sections, the ELF format uses an extended numbering scheme, which is currently not implemented in the I64 linker. That limits the number of sections a single input object module or shareable image can have. Because the linker usually creates the shareable images with sections, this limit applies for creating shareable images as well. Here, ELF sections are used for exporting C++ templates or shared sections. That is the maximum number of such interfaces in a shareable image must be fewer than 65280.

5.26.13 Incorrect Creation Date of Shareable Images in the Map File

V8.2

On OpenVMS I64 platforms, shareable images might show an incorrect creation date in the linker map. The incorrect date is shown is usually 3686. This condition occurs when the linker processes the shareable image as an input file and then extracts the date field, which is shown in the map. The image itself has the correct date that you can see from ANALYZE/IMAGE output.

This problem will be fixed in a future release of the I64 Linker.

5.27 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.

5.28 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.

5.29 OpenVMS Debugger

The following release notes describe conditions specific to the OpenVMS Debugger on OpenVMS I64 and Alpha systems.

For release notes about the DELTA and XDELTA debuggers, see Section 5.13.

5.29.1 STEP/SEMANTIC_EVENT Information Missing in Debugger Command-Line Help

V8.3

Information for the /SEMANTIC_EVENT qualifier does not display properly in Debugger command-line help under the `STEP` command. Information for this qualifier can be found in the command reference section of the *HP OpenVMS Debugger Manual*, under `STEP` command.

5.29.2 Problems and Conditions Corrected in this Release (I64 Only)

V8.3

The following list describes conditions that are corrected in this release:

General Condition: Arrays whose elements are on unaligned boundaries formerly did not displayed properly. This occurred when a programming language construct was used to force unnatural data alignment (for example, `#pragma nomember_align` in C or the `PACKED` keyword in Pascal). This condition is fixed.

COBOL Language Condition: Previously, data declared with EDIT picture clauses were only partially supported. This condition is fixed.

Fortran Language Condition: Previously, a deposit of a value into a data item declared as REAL*16 resulted in a zero being stored rather than the actual value. This condition is fixed.

Pascal Language Condition: Previously, arrays declared with dynamic bounds (for example, "array [1..n]") were not always handled correctly. This condition is fixed.

5.29.3 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 5.10 for additional details.

Note

This is a rare condition that occurs only 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.

5.29.4 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.

5.29.5 C++ Language Issues (I64 Only)

V8.2

Condition: The SHOW CALLS command sometimes displays C++ mangled names, rather than unmangled names.

Workaround: None.

5.29.6 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.

5.29.7 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.

5.29.8 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.

5.29.9 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.

5.29.10 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).

5.29.11 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.

5.30 OpenVMS System Dump Analyzer (SDA)

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

5.30.1 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

5.31 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 2.6.

5.32 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.

5.32.1 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.

5.32.2 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.

5.32.3 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.

5.32.4 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.

5.32.5 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.

5.32.6 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.

5.32.7 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.

5.32.8 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.

5.33 RTL Library (LIB$)

The following notes pertain to the LIB$ run-time library.

5.33.1 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.

5.33.2 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.

5.34 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.

5.35 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.

5.35.1 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.

5.35.2 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.

5.35.3 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")))
```

5.35.4 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).

5.35.5 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.

5.36 System Services

The following notes pertain to OpenVMS system services.

5.37 Timer Queue Entries (TQEs)

Permanent Restriction

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 (TQE$L_ TQFL/TQE$L_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.

5.38 Watchpoint Utility (I64 Only)

V8.2

The Watchpoint Utility has not been ported to OpenVMS I64. HP intends to port this utility in a future release.

5.39 Whole Program Floating-Point Mode (I64 Only)

V8.3

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`

5.40 HP OpenVMS Debugger Heap Analyzer Conditions and Workarounds (I64 only)

V8.2-1

Condition: The Heap Analyzer startup and the Heap Analyzer START command on the I64 kept debugger (START HEAP_ANALYZER) hangs for threaded applications with upcalls enabled.

Workaround: For threaded or AST-involved applications, HP strongly recommends that you either start the Heap Analyzer *before* setting up any debug events or *after* disabling or canceling those events. (You can enable/reset events after the Heap Analyzer startup and START returns you to debugger control.)

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)
- sx1000 Integrity Superdome (Section 6.18)
- ZLX graphics boards (Section 6.19)

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 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*.

6.4.3 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.

6.4.4 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.

6.4.5 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 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.

6.9.3 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.

6.9.4 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.

6.9.5 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.

6.9.6 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
```

6.9.7 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.

6.9.8 Boot Reset Recommendation (Alpha Only)

HP recommends that the console variable `boot_reset` be set to ON.

6.9.9 No Overlay Planes

Hardware overlay planes are not supported.

6.9.10 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 6.9.16.)

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.

6.9.11 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.

6.9.12 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.

6.9.13 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.

6.9.14 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.

6.9.15 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.

6.9.16 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.

6.9.17 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 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.

6.10.2 Multiple USB Keyboards and Mice Not Supported on DECwindows on OpenVMS V8.3

V8.3

DECwindows supports input from only a single mouse or keyboard. If additional keyboards or mice are connected when the system is booted, only the devices configured as KBD0 and MOU0 will function, the others will be inoperable. If additional keyboards or mice are connected after DECwindows has been started, they can cause a system crash. The system crash will be corrected in an ECO patch subsequent to this release.

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 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 with an Intel Saturn I/O (SIO) 82378 chip in your configuration. You must use a SCSI CD, 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.

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.

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.

The RF_VERS utility, a utility program that displays the microcode revision level of the DSSI disk devices, is also provided on the CD. 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

(continued on next page)

Table 6–3 (Cont.) Commands for Updating Microcode in Certain DSSI Disk Devices

Device Type	Platform	Command
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 sx1000 Integrity Superdome

V8.3

The HP Integrity Superdome cannot boot as a satellite over the Core I/O LAN card. When you specify the LAN card as a boot option to BOOT_OPTION.COM and you then shut down the operating system, the LAN card does not appear in EFI. The problem will be fixed in a future release of the firmware.

6.19 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.

6.20 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 5.4.

6.20.1 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 6.20.2.

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.

6.20.2 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 6.20.1.)

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*.

6.21 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.

6.22 Possible Per-Threads Security Impact on Alpha Device Drivers

V7.2

See Section 5.4.7 for information about how possible per-thread security impacts OpenVMS Alpha device drivers.

6.23 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.

6.24 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.

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.

6.25 Adapter Release Notes

V8.2-1

The following sections provide release notes for adapters supported with OpenVMS Version 8.2–1.

6.25.1 Fibre Channel EFI Driver and Firmware Requirements

OpenVMS Version 8.3 for Integrity servers requires that the HP A6826A 2 GB Fibre Channel host-based adapter and its variants have the following minimum version: EFI driver: 1.47 RISC firmware: 3.03.154; HP AB378A and AB379A 4 GB Fibre Channel host-based adapter have the following minimum version: EFI driver: 1.05 RISC firmware: 4.00.70.

To determine the latest, currently supported versions of the RISC firmware and EFI driver, see the README text file provided on the HP IPF Offline Diagnostics and Utilities CD. To locate this file, navigate to the (\efi\hp\tools\io_ cards\fc2) directory for the 2 GB Fibre Channel HBA or \efi\hp\tools\io_ cards\fc4 for the 4 GB HBA. To update the driver and firmware, execute the `fcd_update2.nsh` or the `fcd_update4.nsh`, depending on your HBA type. Instructions for obtaining the Offline Diagnostics and Utilities CD are included in the *HP OpenVMS Version 8.3 Upgrade and Installation Manual*.

6.25.2 Booting with Multiple Fibre Channel Boot Entries

On cell-based systems and newer entry-level systems, the first fibre channel boot entry list is the only valid boot entry. To boot from the other Fibre Channel I64 system disk, go to the EFI Shell and execute "search all", exit the EFI Shell, then select the specified boot entry. This is also required when booting multi-member shadowed system disk.

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 web site 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 6.19 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://h18000.www1.hp.com/support/svctools/`

- **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's `CONVERT` command 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 NetBeans Version 3.6 Support Ends with OpenVMS Version 8.3

V8.3

OpenVMS Alpha and I64 Version 8.3 are the last releases on which NetBeans Version 3.6 for OpenVMS is supported. NetBeans Version 3.6 will be supported over the support life of OpenVMS Version 8.3. Also, note that NetBeans Version 3.6 for OpenVMS Alpha and I64 is supported only on Java™ Platform, Standard Edition, Development Kit (JDK) v 1.4.2-x.

A.9 NetBeans Version 3.6 Support Ends with OpenVMS Version 8.3

For a GUI-based development environment on OpenVMS, please consider using Distributed NetBeans, which provides a cost-effective and flexible development environment solution. More information about Distributed NetBeans can be found at:

`http://www.hp.com/products/openvms/distributednetbeans/`

A.10 Last Ordering Dates for Alpha Systems

V8.2

The following deadlines apply for ordering Alpha systems:

Product	Final Order Date
Alpha systems	October 27, 2006
CPU upgrades	November 30, 2007

A.11 Archived Manuals

V8.3

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.

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:

```
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

64-bit math functions, 5–10
64-bit sigaction problem fixed, 5–10
2-GB malloc, 5–10

A

Ada compiler, 5–8
adapter release notes, 6–22
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 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 utility (I64), 5–24
  releasing process resources fixed, 5–24
  selective output for debug fixed, 5–24
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 6–13
  DECW$OPENGLSHR_RADEON.EXE renamed, 6–9
  DECwindows server hangs, 6–10
ATI RADEON 7500 graphics (cont'd)
  OpenGL supports IEEE arithmetic only, 6–10
  video artifacts at high refresh rates, 6–10
atof problem fixed, 5–14
AUDIT_SERVER
  fails to initiate, 3–10

B

Backport library, 5–11
Backup API
  journaling events, 5–8
Bitmap memory requirements for volume shadowing, 3–8
BLISS
  See HP BLISS compiler
BMC console restrictions (I64 only), 6–1
BUGCHECKFATAL system parameter, 5–16
<builtins.h> changes, 5–13

C

C compiler
  See HP C compiler
C programs
  compiling and case sensitivity, 5–8
C RTL, 5–9 to 5–14
  64-bit math functions, 5–10
  atof("NAN") fixed, 5–14
  backport library, 5–11
  <builtins.h> changes, 5–13
  confstr, 5–11
  DECC$*.OLB libraries frozen, 5–14
  DECC$SHRP.EXE, 5–12
  <decc$types.h> changes, 5–12
  exec* memory leak fixed, 5–10
  _FAST_TOUPPER macro added, 5–14
  __fci builtin added, 5–14
  fcntl, 5–13
  file-pointer-locking hang, 5–11
  fopen, 5–11
  fwrite, 5–13
  2GB malloc, 5–10
  nanosleep, 5–13
  sigaction, 5–10
  socket transfers, 5–13
  struct time_t changes, 5–12

C RTL (cont'd)
TCP/IP header updates, 5–9
<time.h> changes, 5–11
toacii function added, 5–9
<wchar.h> changes, 5–13
C Run-Time Library
See C RTL
C++ compiler
See HP C++ compiler
CANCEL SELECTIVE function, improved use with LTDRIVER, 5–30
CDSA, 5–15
Third party signing process, 1–10
cell-based systems
multiple nPartitions on, 4–14
Circuit switching
and reduced performance, 4–12
CLUE commands
not ported to I64, 5–33
Cluster compatibility kits, 4–10
Clusters
See OpenVMS Cluster systems
CMAP files
new, 2–2
COBOL RTL
See HP COBOL RTL
COM for OpenVMS
error with heavy load of applications, 2–2
support, 2–2
Command Definition utility (I64), 5–24
record attributes fixed for I64 images, 5–25
Common Data Security Architecture
See CDSA
Compaq Open3D layered product
not supported in V8.2, A–1
Compilers
noncompliant code, B–1, B–5
confstr, 5–11
CPUSPINWAIT bugcheck, 5–16
CRCTX routines enhanced, 6–21
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, 5–28
DCE RPC
See HP DCE RPC for OpenVMS
DCL command procedures
using SET SHADOW and SHOW SHADOW, 4–20
DCL commands, 3–2
DDB structure
updates, 5–4
DDT Intercept Establisher Routines
device configuration, 4–12
Debugger
See OpenVMS Debugger
DEC PL/I, 2–3
DECamds
not supported on V8.2, A–2
DECC$*.OLB libraries frozen, 5–14
DECC$SHRP.EXE, 5–12
<decc$types.h> changes, 5–12
DECdfs for OpenVMS
Version 2.4 required, 2–2
DECdtm
Oracle 8*i*, 9*i*, 4–3
DECevent, A–3
not supported, A–2
DECmigrate
not on V8.2 Open Source Tools CD, 3–2
DECnet for OpenVMS, 1–3
DECnet-Plus for OpenVMS, 1–3
new version required, 1–13
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, 6–13
graphics boards support, 6–20
peripheral connection requirements, 1–8
DECwrite
support ending, A–2
Delete key
requires remapping (I64 only), 6–2
Delta/XDelta debuggers, 5–16
register display considerations, 5–16
Device configuration, 4–12
Device drivers
IPL setup, 6–21
MON version handling, 6–21
per-thread security impact, 6–21
recompiling and relinking, 6–20 to 6–21
SCSI, 6–20
DEVICE_NAMING system parameter, 4–4
DIAGNOSE command
not supported, 3–2, A–2
DIGITAL Modular Computing Components (DMCC)
KZPDA controller and PBXGA graphics card, 6–13
updating the SRM console, 6–13

Digital Personal Workstation, 6–14
Dissimilar device shadowing (DDS)
  KZPDC restriction, 4–21
  Smart Array 5300 restriction, 4–21
  write bitmap interaction, 4–20
Documentation changes and corrections
  archived manuals, A–4
  Guide to OpenVMS File Applications, 5–16
  LIB$ help omission, 5–37
  OpenVMS Performance Management, 4–17
  OpenVMS RTL Screen Management (SMG$) Manual, 5–38
Documentation corrections, 3–3
  $PUTMSG system service, 3–8
  using IPC commands, 3–7
DSSI disk devices
  microcode revision levels, 6–16
Dual-controller HSG*nn*
  failure, 6–14
Dynamic CPU configuration
  POSIX Threads Library, 5–36

E

ECP (Enterprise Capacity and Performance), 4–4
EDIT/FDL
  fixing recommended bucket size, 4–4
EFI driver, 6–22
EFI precautions, 4–5
EFI$CP utility
  use not recommended, 4–4
ELV
  See Error Log Viewer (ELV) utility
Enterprise Capacity and Performance (ECP)
  See ECP
Error Log Report Formatter (ERF)
  unsupported, A–3
Error Log Viewer (ELV) utility, 4–6
EV6 Alpha processor, B–1
`exec*` memory leak fixed, 5–10
Extended DDT bit
  problem corrected, 5–30
Extended File Cache (XFC), 4–19
External authentication, 4–7
  I64 support, 4–7
  password expiration notification, 4–7
  SET PASSWORD command, 4–7

F

Fast Path
  turning off, for Galaxy on ES40, 4–15
_FAST_TOUPPER macro added, 5–14
`__fci` builtin added, 5–14
`fcntl`, 5–13
Fibre Channel, 6–22
  compatibility kits, 4–10
  multipath failover restriction for tape devices, 4–13
File attributes
  updated recommendations, 4–2
File-pointer-locking hang, 5–11
Firmware
  for Alpha servers, 1–8
  for Integrity servers, 1–6
Floating-point data
  considerations for applications, 5–6
FMS kits, 2–3
`fopen`, 5–11
Fortran
  See HP Fortran
Freeware, 3–1, A–1
  menu unavailable on OpenVMS I64, 3–1
`fwrite`, 5–13

G

Galaxy
  definitions, 4–14
gigabit Ethernet controller
  restriction, 1–4
Graphical Configuration Manager (GCM)
  supported for Galaxy, 4–15
Graphical Configuration Utility (GCU), 4–15
Graphics
  support for I64 systems, 6–8
Graphics boards support, 6–20

H

Hard partition, 4–14
HP BLISS compiler
  consequences of noncompliant code, B–1
  warnings (I64 only), 5–17
HP C compiler
  consequences of noncompliant code, B–1
HP C++ compiler
  consequences of noncompliant code, B–1
HP COBOL RTL, 5–18
HP DCE RPC for OpenVMS, 2–4 to 2–6
HP DECram, 2–6
  command conflict between DECRAM and DECRYPT, 2–7
  removal before upgrade (Alpha only), 1–13
  ships as a SIP on V8.2, 2–7
  Version 2.5 (VAX only), 2–7
HP DECwindows Motif
  keyboard support restrictions (I64), 1–7
  pause screen can fail to unlock (Alpha only), 2–7
  startup messages, 1–8
  user-written transport support, 2–7
  version support, 1–4

HP Fortran
for I64, 5–18
HP MACRO for OpenVMS, 5–19
floating divide-by-zero error not raised (I64 only), 5–21
on Alpha systems, 5–20
on I64 systems, 5–19
/OPTIMIZE=VAXREGS qualifier not supported on I64, 5–21
/TIE qualifier defaults on Alpha and I64, 5–21
HP Secure Web Browser
increased memory requirement, 3–2
installation error on ODS-2 (I64 only), 3–2
HP Secure Web Server
support, 2–8
HP SSL
Startup commands for Encrypt and SSL, 1–10
HSG*nn*
failure, 6–14
Hypersort utility, 5–21 to 5–23

I

IDE CD, 6–14
IEE Floating Point
filter (I64 only), 5–6
INIT console command
usage on ES47/ES80/GS1280 soft partitions, 6–3
Installation and upgrade information
networking options, 1–3
Installation error
HP Secure Web Browser, 3–2
Integrity server
firmware, 1–6
Intel® Assembler (I64 only), 5–23
Interlocked memory instructions, B–1
Invocation context block, 5–37
IPC Commands, 3–7
ISA_CONFIG.DAT
unsupported in future release, A–3

K

Kerberos
Default startup order required for OpenVMS and Kerberos ACME agents, 4–24
Kerberos for OpenVMS, 1–10
KPB extensions, 5–3

L

LAN Drivers
duplex mode mismatch errors, 3–9
LANCP
converting device database after upgrading (Alpha only), 1–13
Layered product
fails to install, 1–15
LIB$GET_CURR_INVO_CONTEXT
documentation correction, 3–7
LIB$GET_INVO_CONTEXT
documentation correction, 3–7
LIB$GET_INVO_HANDLE
documentation correction, 3–7
LIB$GET_PREV_INVO_CONTEXT
documentation correction, 3–7
LIB$GET_PREV_INVO_HANDLE
documentation correction, 3–7
LIB$GET_UIB_INFO
documentation correction, 3–7
LIB$I64_GET_FR, 5–37
LIB$I64_GET_GR, 5–37
LIB$I64_PUT_INVO_REGISTERS, 5–37
LIB$I64_SET_FR, 5–37
LIB$I64_SET_GR, 5–37
LIB$LOCK_IMAGE
missing from help, 5–37
LIB$PUT_INVO_REGISTERS
documentation correction, 3–7
LIBRARIAN
See Librarian Utility, 3–5
Librarian utility, 3–5, 5–23
error reporting problem, 5–24
linking against data-reduced ELF object libraries (I64 restriction), 5–23
restrictions with .STB files (I64 only), 5–23
Library utility
corrected information
/accessing ELF object libraries, 3–5
/REMOVE, 3–5
LIBRTL
Calling Standard routines (I64 only), 5–37
Licenses, 4–10
Licensing issues, 6–5 to 6–7
Linker for OpenVMS Alpha, 5–25 to 5–26
change in behavior with library check, 5–25
hangs when processing many files, 5–25
limit of 25 elements on stack, 5–26
RMS_RELATED_CONTEXT option, 5–25
Linker for OpenVMS I64, 5–26 to 5–29
created code stubs, 5–29
data-reduced ELF object libraries, 5–28
demangled symbol names, 5–29
differences from OpenVMS Alpha Linker, 5–28
/EXPORT_SYMBOL_VECTOR removed, 5–29
initialized overlaid program sections, 5–29
LINK_ORDER section header flag, 5–28
longer symbol names in options, 5–29
/PUBLISH_GLOBAL_SYMBOLS removed, 5–29
LINK_ORDER ELF section header flag, 5–28
LTDRIVER restriction, 5–30

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, 5–30
Microcode revision levels
  commands for updating, 6–17
  on DSSI disk devices, 6–16
Migration software, 1–15
MMG_CTLFLAGS system parameter, 4–17
Monitor utility changes, 4–1
MP console restrictions (I64 only), 6–1
Multipath failover
  Fibre Channel tape device restriction, 4–13
  tape robots, 4–13
multiple nPartitions
  on cell-based systems, 4–14
MULTIPROCESSING system parameter, 5–16

N

`nanosleep`, 5–13
NetBeans
  NetBeans V3.6 support ends with OpenVMS V8.3, A–3
  Requires Java Standard Edition, Development Kit v 1.4.2-x, 2–9
Network
  update restrictions, 3–8
Network options, 1–3
Noncompliant code, B–1, B–2

O

Open3D graphics
  controller boards support, 6–20
  licensing change, 6–14
OpenVMS
  ENCRYPT and DECRYPT commands, 1–12
OpenVMS Calling Standard
  rotating registers, 5–14
OpenVMS Cluster systems, 4–8 to 4–13
  compatibility kits, 4–10
  compatibility kits for mixed versions, 4–10
  patch kits, 4–9
  performance reduced with CI-LAN switching, 4–12
  rolling upgrades, 1–9
  SCSI multipath failover, 4–11
OpenVMS Debugger, 5–30, 5–41
  Ada event support, 5–7
  Alpha
    previous versions not supported, 5–33
  Alpha and I64
    change to SET SCOPE command, 5–33
    change to SHOW IMAGE command, 5–33
  C++ language issues, 5–8
  conditions corrected, 5–31
  Ctrl/C and STOP limitation (Alpha only), 5–7
  EXAMINE/INSTRUCTION %PREVLOC command fixed, 5–7
  Heap Analyzer conditions, 5–41
  I64
    BASIC language issues, 5–32
    C++ language issues, 5–32
    COBOL language issues, 5–32
    Fortran language issues, 5–32
    general conditions and workarounds, 5–31
    Pascal language issues, 5–32
  PC client documentation correction, 3–3
  SHOW MODULE command enhancement, 5–8
  SHOW SYMBOL/TYPE behavior correction, 5–7
OpenVMS DELTA/XDELTA Debugger
  update to manual, 3–4
OpenVMS Galaxy, 4–14 to 4–15
  and ES40
    turning off Fast Path, 4–15
    uncompressed dump limitation, 4–15
  license enforcement, 6–5
OpenVMS I64
  booting from DVD, 1–7
OpenVMS Management Station, 4–15
OpenVMS Performance Management
  documentation correction, 4–17
OpenVMS Registry
  Version 2 format database corruption, 4–15
OpenVMS System Dump Analyzer
  CLUE commands not ported to I64, 5–33
OpenVMS Utility Routines manual
  update, 3–4

P

Partition
  hard, 4–14
  soft, 4–14
Pascal
  reinstalling after an upgrade (Alpha), 2–9
  V5.8A required to create STARLET library (Alpha only), 2–8
Patch kits
  required for mixed-version OpenVMS Cluster system, 4–10
Patch kits for cluster compatibility, 4–9

PCB$T_TERMINAL
size increase, 5–4
PCI configuration restriction, 6–8
PEdriver
response to LAN congestion, 4–12
Per-thread security
impact on device drivers, 5–5
impact on privileged code, 5–5
PGFLQUOTA problems, 5–24
PL/I
libraries not included in I64, 5–34
RTL support, 2–3
POOLCHECK system parameter, 5–16
Port driver $QIO
restriction, 5–30
POSIX Threads Library, 5–34 to 5–37
debugger metering does not work, 5–37
dynamic CPU configuration, 5–36
floating-point exceptions (I64 only), 5–35
POSIX 1003.4a, Draft 4 interface to be retired, A–3
stack overflows during exception handling (I64 only), 5–34
THREADCP command behavior for I64, 5–35
PowerStorm 300/350 PCI graphics support, 6–15
Open3D license no longer checked, 6–15
Privileged data structures
64-bit logical block number, 5–3
CPU name space, 5–3
forking to a dynamic spinlock, 5–3
KPB extensions, 5–3
PCB$T_TERMINAL size increase, 5–4
per-thread security impact on, 5–5
UCB/DDB updates, 5–4
updates to, 5–2 to 5–4
Programming
$PUTMSG system service correction, 3–8
PTD$ pseudoterminal driver
corrected information /REMOVE, 3–5
PTD$READ routine
documentation clarification, 3–4

R

Recompiling programs
for Alpha, 5–2
Remedial kits
obtaining, 1–3
required for mixed-version OpenVMS Cluster system, 4–10
Retired products information, A–1 to A–4
RF73 and RF*nn* disks, controller memory errors, 6–16
RMS
FAB, 5–18
Rotating registers, 5–37
Run-time library routines, 3–7
rx7620 server, 1–4
rx8620 server, 1–4
RZ*nn* disk drives, 6–18 to 6–19

S

SCSI controllers
restrictions on AlphaServer 2100 systems, 6–3
SCSI device drivers, 6–20
SCSI multipath incompatibility, 4–11
SDA
See OpenVMS System Dump Analyzer
sections, 5–30
Security
file attributes, 4–2
Servers
rx7620, 1–4
rx8620, 1–4
Superdome, 1–4
SET DEVICE/SWITCH command, 4–13
SET PASSWORD command, 4–7
SET SHADOW
using in DCL command procedures, 4–20
SHADOW_MAX_UNIT
default setting, 1–14
SHAD_MAX_UNIT
memory consumption, 1–14
SHOW SHADOW
using in DCL command procedures, 4–20
`sigaction`, 5–10
Signal array
missing on I64, 4–25
Smart Array 5300
Volume Shadowing restriction, 4–21
SMG$
documentation corrections, 5–38
Socket transfers, 5–13
Software Public Rollout Reports, 2–1
SORT32 utility, 5–23, 5–39 to 5–40
SPLINVIPL bugcheck, 5–6
SRM_CHECK tool, B–1
`struct time_t` changes, 5–12
Superdome
sx1000, 6–19
Superdome server, 1–4
Support policy for software, 1–1
sx1000 Chipset, 1–4
sx1000 Superdome, 6–19
System crashes
recovery from (I64 only), 4–3
System disk
incompatibility with older systems, 1–3
System Event Analyzer (SEA) utility
support on I64, 2–9

System Event Log (SEL)
clearing on Integrity servers, 1–5
System hangs
recovery from (I64 only), 4–3
System parameters, 4–16 to 4–17
BUGCHECKFATAL, 5–16
changes, 4–17
DEVICE_NAMING
used to increase device unit number maximum, 4–4
MMG_CTLFLAGS documentation error, 4–17
MULTIPROCESSING, 5–16
new parameters, 4–16
obsolete parameters, 4–16
POOLCHECK, 5–16
SYSTEM_CHECK, 5–16
System service changes, 5–1
SYSTEM_CHECK system parameter, 5–16

T

Tape robots
automatic multipath failover, 4–13
TCP/IP header updates, 5–9
TCP/IP Services for OpenVMS, 1–3
Terminal Fallback Facility (TFF), 4–17
restrictions, 4–18
TFF
See Terminal Fallback Facility
THREADCP command
behavior for I64, 5–35
TIE kit, 1–15
<time.h> changes, 5–11
Timer queue entries (TQEs), 5–40
toacii function added, 5–9
TQEs
See Timer queue entries
Traceback facility
API problem fixed, 4–24
Translated Image Environment
See TIE kit, 1–15

U

UCB structure
updates, 5–4
UI60 SCSI Support
rx7620, rx8620, 1–5
Upgrade
paths, 1–9
USB
keyboards and mice not supported on DECwindows, 6–13

V

VAX Cluster Cache
See Virtual I/O Cache
VCC
See Virtual I/O Cache
VIOC
See Virtual I/O Cache
Virtual I/O Cache
not available on I64, 4–19
superseded by XFC, 4–19
Volume Shadowing for OpenVMS, 4–19 to 4–23
compatibility kits, 4–10
device name requirement, 4–19
dissimilar device shadowing (DDS)
caution, 4–20
KZPDC restriction, 4–21
EFI precautions with shadowed system disk, 4–5
memory requirements for bitmaps, 3–8
performance of merge operations, 4–22
problem with dismount using /MINICOPY, 4–22
Smart Array 5300 (KZPDC) restriction, 4–21

W

Watchpoint utility, 5–40
<wchar.h> changes, 5–13
WEBES
support on I64, 2–9
Whole-program floating-point mode (I64 only), 5–40
Write bitmaps
and dissimilar device shadowing (DDS), 4–20

X

X.25 data links unsupported (I64 only), 2–6
XA, 4–3
XFC
See Extended File Cache

Z

ZLX graphics boards support, 6–20