# HP OpenVMS Version 8.4 Release Notes

**June 2010**

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 Version 8.4 for Integrity servers<br>OpenVMS Alpha Version 8.4 |

**Hewlett-Packard Company**
**Palo Alto, California**

© Copyright 2010 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 and Windows are U.S. 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

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 OpenVMS Integrity server Users . . . . . 1–4
1.7.1 Storage Controllers . . . . . 1–4
1.7.2 U160 SCSI Support for rx7620 and rx8620 . . . . . 1–5
1.7.3 Clearing the System Event Log 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 Booting from USB or vMedia Devices . . . . . 1–8
1.7.7 HP DECwindows Motif Release Notes . . . . . 1–8
1.7.7.1 Keyboard Support . . . . . 1–8
1.8 OpenVMS Alpha Users . . . . . 1–8
1.8.1 Firmware for OpenVMS Alpha Version 8.4 . . . . . 1–8
1.8.2 Upgrade Paths . . . . . 1–8
1.9 Kerberos for OpenVMS . . . . . 1–10
1.10 Modifying SYSTARTUP_VMS.COM . . . . . 1–12
1.11 Encryption for OpenVMS . . . . . 1–12
1.12 Upgrading HP DECram V3.*n* . . . . . 1–12
1.13 Converting the LANCP Device Database . . . . . 1–13
1.14 DECnet-Plus Requires a New Version . . . . . 1–13
1.15 Remove TIE Kit Before Upgrade . . . . . 1–14
1.16 Installation Failure of Layered Products on Alternate Devices or Directories . . . . . 1–14

## 2 OpenVMS Associated Products Release Notes

2.1 Associated Product Support . . . . . 2–1
2.2 HP TCP/IP Services for OpenVMS . . . . . 2–2
2.3 NetBeans Version 5.5.1 Requires Latest JDK . . . . . 2–2
2.4 Problem Accessing DFS Mounted Disk . . . . . 2–2
2.5 HP DCE for OpenVMS Restriction (Integrity servers Only) . . . . . 2–2
2.6 XML-C Product Zip File . . . . . 2–3
2.7 OpenVMS e-Business and Integration Infrastructure Package . . . . . 2–3
2.8 Updates to Freeware Readme File . . . . . 2–3
2.9 CMAP Files Added . . . . . 2–4

2.10 COBOL: Changes in I/O Run-Time Diagnostics and RMS Special Registers . . . 2–4
2.11 COM for HP OpenVMS (Alpha Only) . . . 2–4
2.11.1 COM for OpenVMS Support . . . 2–4
2.11.2 Registry Access Error with Heavy Load of Applications . . . 2–4
2.12 DECdfs Version 2.4 Required for OpenVMS Version 8.3 . . . 2–4
2.13 DECforms Web Connector Version 3.0 (Alpha Only) . . . 2–5
2.14 DEC PL/I: RTL Support for OpenVMS . . . 2–5
2.15 FMS Kits . . . 2–5
2.16 Graphical Configuration Manager (Alpha Only) . . . 2–6
2.17 HP DECram . . . 2–6
2.17.1 DECram Available with OpenVMS Version 8.2 and later . . . 2–6
2.17.2 Conflict with DECRYPT DCL Command . . . 2–6
2.18 HP DECwindows Motif for OpenVMS . . . 2–6
2.18.1 New Locales Added . . . 2–7
2.18.2 User-Written Transports not Supported . . . 2–7
2.19 HP Secure Web Server Version Support . . . 2–7
2.20 HP Pascal for OpenVMS Alpha Systems . . . 2–7
2.20.1 HP Pascal: Version 5.8A (or later) Required to Create STARLET Library (Alpha Only) . . . 2–8
2.20.2 Installing HP Pascal After an Upgrade (Alpha Only) . . . 2–8
2.21 WEBES and SEA Support on Integrity servers . . . 2–8

## 3 General User Release Notes

3.1 SYS$GETTIM_PREC System Service Declaration . . . 3–1
3.2 Problem With F$GETSYI("RAD_CPUS") . . . 3–1
3.3 HP Code Signing Service for OpenVMS Support . . . 3–1
3.4 Symbolic Links Implementation Changes . . . 3–1
3.4.1 Logical Names . . . 3–1
3.4.2 Audit Alarms Fixed . . . 3–2
3.5 SHOW SYSTEM/STATE=MUTEX Does not Display the Processes . . . 3–2
3.6 SWB V1.1-12 Installation Warnings . . . 3–3
3.7 Ctrl/P at the Console Does not Always Work . . . 3–3
3.8 Installing Oracle Database 10g Release 2 for OpenVMS Fails . . . 3–4
3.9 Serial Port Enumeration . . . 3–4
3.10 Old Firmware Cannot Translate Messages Written to the System Event Log . . . 3–6
3.11 TZ Function in C RTL . . . 3–7
3.12 InfoServer Utility and FDDI . . . 3–7
3.13 New Qualifier for DCL Command SET PASSWORD . . . 3–7
3.14 OpenVMS Freeware CDs . . . 3–7
3.14.1 Freeware Menu Unavailable (Integrity servers Only) . . . 3–8
3.15 DCL Commands . . . 3–8
3.15.1 SHUTDOWN.COM on OpenVMS Graphics Console (Integrity servers only) . . . 3–8
3.15.2 DIAGNOSE Command No Longer Supported . . . 3–8
3.15.3 MOUNT Command Restriction . . . 3–8
3.16 DECmigrate Not on Open Source Tools CD . . . 3–9
3.17 HP Secure Web Browser . . . 3–9
3.17.1 Increased Memory Required . . . 3–9
3.17.2 Installation Error on ODS-2 Disk Volume (Integrity servers Only) . . . 3–9
3.18 Documentation Corrections . . . 3–10

3.18.1 *HP OpenVMS Linker Utility Manual* Update . . . . . 3–10
3.18.1.1 HP C++ Examples . . . . . 3–10
3.18.2 *HP PCSI Utility Online help and Manual*: $PRODUCT REGISTER VOLUME Syntax Error Correction . . . . . 3–10
3.18.3 iCAP Release Notes: GiCAP Functionality not Available . . . . . 3–10
3.18.4 *POLYCENTER Software Installation Utility Developer's Guide*: PRODUCT Command Update . . . . . 3–11
3.18.5 *HP OpenVMS System Manager's Manual Update* . . . . . 3–11
3.18.5.1 Getting Information About Devices on the System . . . . . 3–11
3.18.5.2 Initializing a New Volume with ODS-5 Format . . . . . 3–13
3.18.5.3 Converting from ODS-2 to ODS-5 . . . . . 3–13
3.18.5.4 New Extended File Specifications Characteristics . . . . . 3–14
3.18.5.5 ODS-2 and ODS-5 Used Together . . . . . 3–14
3.18.5.6 Performing Image Backups to Disk . . . . . 3–15
3.18.5.7 Mounting a Volume With Caching Disabled . . . . . 3–16
3.18.5.8 System-Wide Statistics . . . . . 3–16
3.18.5.9 Disabling Caching for a Volume . . . . . 3–16
3.18.5.10 Understanding File System Data Caches . . . . . 3–17
3.18.6 HP OpenVMS RTL Library (LIB$) Manual . . . . . 3–17
3.18.7 Documentation Error: LCKMGR_CPUID System Parameter . . . . . 3–18
3.18.8 MMG_CTLFLAGS: Documentation Error . . . . . 3–18
3.18.9 *HP OpenVMS System Analysis Tools Manual* . . . . . 3–18
3.18.10 *HP OpenVMS Programming Concepts Manual* . . . . . 3–18
3.18.10.1 Saving System Dumps . . . . . 3–18
3.19 *HP OpenVMS DELTA/XDELTA Debugger Manual* Update . . . . . 3–18
3.19.1 *HP OpenVMS Version 8.2 New Features and Documentation Overview*: Librarian Utility Corrections . . . . . 3–18
3.19.1.1 /REMOVE Qualifier Correction . . . . . 3–19
3.19.1.2 Accessing ELF Object Libraries Correction . . . . . 3–19
3.19.2 *HP OpenVMS RTL Library (LIB$) Manual* Corrections . . . . . 3–20
3.19.2.1 *HP OpenVMS RTL Library (LIB$) Manual*: Rounding Rule for LIB$CVT_DX_DX . . . . . 3–20
3.19.3 *HP OpenVMS RTL Library (LIB$) Manual*: Platform Restrictions . . . 3–20
3.19.4 *HP OpenVMS System Manager's Manual*: IPC Commands Restriction . . . . . 3–21
3.20 Network Update Restrictions from Version 8.2 to Version 8.2–1 . . . . . 3–21
3.21 Synchronous Data Links not Supported . . . . . 3–21
3.22 Duplex-Mode Mismatch Errors . . . . . 3–22

## 4 System Management Release Notes

4.1 SYS$TIMEZONE_RULE Logical Replaces Hyphen (-) with Caret (^) . . . . 4–1
4.2 Issues with Time Zone Configuration . . . . . 4–1
4.3 OpenVMS as a Guest Operating System on Integrity VM . . . . . 4–2
4.3.1 "Guest Punishment" Scenarios . . . . . 4–2
4.3.2 Increased CPU Consumption After Shutdown . . . . . 4–4
4.3.3 OpenVMS Guest Does not Support Attached I/O Devices . . . . . 4–4
4.3.4 Networking or Storage Interface Support . . . . . 4–4
4.4 Provisioning OpenVMS Using HP SIM . . . . . 4–4
4.4.1 Provisioning OpenVMS Guest Limitation . . . . . 4–4
4.4.2 System Firmware . . . . . 4–5
4.4.3 Provisioning Multiple Servers . . . . . 4–5
4.4.4 Provisioning From HP SIM Central Management Server . . . . . 4–5
4.4.5 InfoServer Name Length . . . . . 4–5

4.4.6 OpenVMS InfoServer and the Integrity servers on the Same LAN . . . 4–5
4.4.7 EFI Firmware . . . 4–5
4.4.8 Management Processor . . . 4–5
4.4.9 OpenVMS TCP/IP Provisioning Limitation . . . 4–5
4.4.10 Limitation with Deploying OpenVMS on Multiple Target Servers Simultaneously . . . 4–6
4.5 Insight Dynamics - Virtual Server Environment for OpenVMS . . . 4–6
4.5.1 Utilization Data Collection Fails . . . 4–6
4.5.2 Problem While Creating a New or Replacement Simulated System . . . 4–7
4.5.3 Utilization Data not Available for OpenVMS Sub-OS Workloads . . . 4–7
4.5.4 Insight Software Features not Supported on OpenVMS . . . 4–7
4.6 Performance Enhancements . . . 4–8
4.6.1 Enhancements to Write Bitmaps . . . 4–8
4.6.1.1 WBM_MSG_INT Parameter Updates . . . 4–8
4.6.1.2 WBM_MSG_UPPER and WBM_MSG_LOWER Parameter Updates . . . 4–8
4.6.1.3 Asynchronous SetBit Messages . . . 4–9
4.6.1.4 Reduced SetBit Messages for Sequential I/O . . . 4–9
4.6.2 Exception Handling Performance Improvements (Integrity servers Only) . . . 4–9
4.6.3 Exception Handling (Integrity servers Only) . . . 4–9
4.6.4 Image Activation (Integrity servers Only) . . . 4–10
4.6.5 Global Section Creation and Deletion . . . 4–10
4.6.6 Dedicated CPU Lock Manager . . . 4–10
4.6.7 Ctrl/T Alignment Faults . . . 4–10
4.7 Error and Warning Messages from ACPI During Boot . . . 4–10
4.8 Large Device Name Support for Accounting Utility . . . 4–10
4.9 PAGED_LAL_SIZE New System Parameter . . . 4–11
4.9.1 Paged Pool Lookaside Lists . . . 4–11
4.10 2 TiB Disk Volume Support Restrictions . . . 4–11
4.11 Configuring SAS Tape Drives . . . 4–12
4.12 External SAS Disk Device Naming . . . 4–12
4.13 External Authentication . . . 4–12
4.13.1 External Authentication and Password Policy . . . 4–12
4.13.2 Integrity servers External Authentication Support . . . 4–13
4.13.3 SET PASSWORD Behavior Within a DECterm Terminal Session . . . 4–13
4.13.4 No Password Expiration Notification on Workstations . . . 4–13
4.13.5 Restriction in ACME_SERVER Process (Integrity servers only) . . . 4–13
4.14 Itanium Primary Bootstrap (IPB) Fails to Find the Valid Dump Devices . . . 4–14
4.15 SHUTDOWN.COM Changes . . . 4–14
4.16 OpenVMS Cluster Systems . . . 4–14
4.16.1 Cluster over IP (IP Cluster Interconnect) . . . 4–14
4.16.1.1 Software Requirements . . . 4–14
4.16.1.2 Integrity servers Satellite Node and Bootserver in the Same LAN . . . 4–14
4.16.1.3 Alpha Satellite Node Requires LAN Channels With Disk Server . . . 4–15
4.16.1.4 IPv6 Support . . . 4–15
4.16.1.5 Dynamic Host Configuration Protocol (DHCP) or Secondary Address Support . . . 4–15
4.16.1.6 Multiple IP Interface Configuration . . . 4–15
4.16.1.7 ifconfig Command Usage . . . 4–15
4.16.1.8 Multiple Gateway Configuration . . . 4–15

4.16.1.9 Block Transfer XMIT Chaining . . . . . 4–15
4.16.1.10 LANCP for Downline Load . . . . . 4–16
4.16.1.11 Duplex Mismatch . . . . . 4–16
4.16.1.12 Configuring a Node During Upgrade . . . . . 4–16
4.16.2 OpenVMS Cluster Support for Integrity VM . . . . . 4–16
4.16.2.1 Cluster Interconnect for OpenVMS Guest . . . . . 4–16
4.16.2.2 MSCP Support for Clusters in Integrity VM Environment . . . . . 4–16
4.16.2.3 Online Migration Support . . . . . 4–16
4.16.3 Mixed Platform Support . . . . . 4–17
4.16.4 Satellite Systems using Port Allocation Class . . . . . 4–17
4.17 Mixed-version Cluster Compatibility of a Six-member Shadowset . . . . . 4–17
4.18 Backward Compatibility of a Six-member Shadowset . . . . . 4–18
4.19 WBEM Services and WBEM Providers for OpenVMS . . . . . 4–18
4.19.1 Increased CPU Consumption With WBEM on OpenVMS Guest . . . . . 4–18
4.19.2 WBEM Providers Support for OpenVMS Guest . . . . . 4–18
4.19.3 Based on OpenPegasus 2.9 . . . . . 4–19
4.19.4 Supports nPartitions and iCAP . . . . . 4–19
4.19.5 Restart cimserver.exe to Unload Providers on OpenVMS . . . . . 4–19
4.19.6 Use Quotes Around Command Line Options . . . . . 4–19
4.20 Writing the System Dump File to an Alternate Disk . . . . . 4–19
4.21 Monitor Utility Changes . . . . . 4–19
4.21.1 Guest Operating System on Integrity VM . . . . . 4–19
4.21.2 Version-to-Version Compatibility of MONITOR Data . . . . . 4–21
4.21.3 Playing Back Data from a Recording File . . . . . 4–21
4.22 System Parameters . . . . . 4–21
4.23 SYS$LDDRIVER Restriction . . . . . 4–22
4.24 CPU_POWER_MGMT Default Value Changed . . . . . 4–22
4.25 Booting A Satellite System with Reserved Memory . . . . . 4–22
4.26 SCACP Error Counter Reports Retransmit Errors . . . . . 4–22
4.27 Virtual Connect . . . . . 4–23
4.27.1 Failover and RECNXINTERVAL . . . . . 4–23
4.28 INITIALIZE/ERASE=INIT Before Using Media . . . . . 4–23
4.29 Performance Data Collector for OpenVMS (TDC) . . . . . 4–23
4.30 Recovering From System Hangs or Crashes (Integrity servers Only) . . . . . 4–23
4.31 DECdtm/XA with Oracle 8*i* and 9*i* (Alpha Only) . . . . . 4–24
4.32 Device Unit Number Increased . . . . . 4–24
4.33 EDIT/FDL: Fixing Recommended Bucket Size . . . . . 4–24
4.34 Using EFI$CP Utility not Recommended . . . . . 4–25
4.35 Error Log Viewer (ELV) Utility: TRANSLATE/PAGE Command . . . . . 4–25
4.36 Cluster Compatibility Patch Kits . . . . . 4–25
4.36.1 Patch Kits Needed for Cluster Compatibility . . . . . 4–26
4.36.2 API to Correct Incompatibility of FC and SCSI Multipath with Some Third-Party Products . . . . . 4–27
4.36.3 DDT Intercept Establisher Routines and Device Configuration Notification Results . . . . . 4–28
4.36.4 Cluster Performance Reduced with CI-LAN Circuit Switching . . . . . 4–28
4.36.5 Multipath Tape Failover Restriction . . . . . 4–29
4.36.6 No Automatic Failover for SCSI Multipath Medium Changers . . . . . 4–29
4.37 OpenVMS Galaxy (Alpha Only) . . . . . 4–29
4.37.1 Galaxy Definitions . . . . . 4–30
4.38 Multiple nPartitions on Cell-based Systems . . . . . 4–30
4.38.1 OpenVMS Graphical Configuration Manager . . . . . 4–30
4.38.2 Galaxy on ES40: Uncompressed Dump Limitation . . . . . 4–30
4.38.3 Galaxy on ES40: Turning Off Fastpath . . . . . 4–31

4.39 Corrupted Version 2 Format Database . . . . . 4–31
4.40 System Parameters . . . . . 4–31
4.40.1 New System Parameters . . . . . 4–31
4.40.2 Obsolete System Parameters . . . . . 4–31
4.40.3 System Parameter Changes . . . . . 4–32
4.41 Terminal Fallback Facility . . . . . 4–32
4.42 User Environment Test Package (Integrity servers Only) . . . . . 4–33
4.43 Recommended Caching Methods . . . . . 4–34
4.44 Analyze Utility for OpenVMS . . . . . 4–34
4.44.1 Formatted Symbol Vector Correctly Shown in Data Segment . . . . . 4–34
4.44.2 Transfer Array Formatted in Data Segment . . . . . 4–34
4.44.3 System Version Array Formatted in Dynamic Segment . . . . . 4–35
4.44.4 Enhancements for the /SEGMENT Qualifier . . . . . 4–35
4.44.5 Support for Section Escaping Added . . . . . 4–35
4.45 INSTALL Utility for OpenVMS (Installing Resident Images in S2 Space) . . . . . 4–35

## 5 Programming Release Notes

5.1 Symbolic Debugger . . . . . 5–1
5.2 C++ Run-Time Library . . . . . 5–1
5.3 Process/Application Hangs . . . . . 5–3
5.4 AST Delivery Clarification in Programs using POSIX Threads . . . . . 5–3
5.5 RMS $PARSE Validation of Directory Files . . . . . 5–4
5.6 No-IOLOCK8 Fibre Channel Port Drivers . . . . . 5–4
5.7 C++ Compiler . . . . . 5–5
5.8 Building DCE IDL C++ Applications . . . . . 5–5
5.9 Privileged Programs may Need a Recompile (Alpha Only) . . . . . 5–6
5.10 Privileged Data Structures Updates . . . . . 5–6
5.10.1 KPB Extensions . . . . . 5–7
5.10.2 CPU Name Space . . . . . 5–7
5.10.3 64-Bit Logical Block Number (LBN) . . . . . 5–7
5.10.4 Forking to a Dynamic Spinlock . . . . . 5–7
5.10.5 UCB/DDB Updates . . . . . 5–8
5.10.6 PCB$T_TERMINAL Size Increase . . . . . 5–8
5.10.7 Per-Thread Security Impacts Privileged Code and Device Drivers . . . . 5–8
5.11 Applications Using Floating-Point Data . . . . . 5–10
5.11.1 IEEE Floating-Point Filter (Integrity servers Only) . . . . . 5–10
5.11.2 Ada Event Support (Integrity servers Only) . . . . . 5–10
5.11.3 C++ Language Issues (Integrity servers Only) . . . . . 5–11
5.12 Ada Compiler(Integrity servers Only) . . . . . 5–11
5.13 Backup API: Journaling Callback Events Restriction . . . . . 5–11
5.14 C Programs: Compiling with CASE_LOOKUP=SENSITIVE Settings . . . . 5–11
5.15 C Run-Time Library . . . . . 5–11
5.15.1 C RTL TCP/IP Header File Updates . . . . . 5–12
5.15.2 Backport Library No Longer Shipped . . . . . 5–12
5.15.3 Header File <time.h> Changes . . . . . 5–12
5.15.4 Header File <time.h> Makes *_r Non-ANSI Functions Visible . . . . . 5–13
5.15.5 Header File <builtins.h> __CMP_SWAP* and _Interlocked* Visible to C++ . . . . . 5–13
5.15.6 Builtin __fci Added for Integrity servers . . . . . 5–13
5.15.7 No New Entries for DECC$*.OLB Object Libraries . . . . . 5–13
5.16 Calling Standard and Rotating Registers (Integrity servers Only) . . . . . 5–13
5.17 Common Data Security Architecture (CDSA) Considerations . . . . . 5–14

5.17.1 Secure Delivery . . . . . 5–14
5.17.2 Installation and Initialization Considerations . . . . . 5–14
5.18 Debugging Modes: Avoiding CPUSPINWAIT Bugchecks . . . . . 5–15
5.19 Delta/XDelta Debuggers . . . . . 5–15
5.19.1 XDELTA Register Display Consideration (Integrity servers Only) . . . . 5–15
5.20 File Applications: Corrections to *Guide to OpenVMS File Applications* . . . . . 5–16
5.21 HP BLISS Compiler Warnings with RMS Structures (Integrity servers Only) . . . . . 5–16
5.22 Potential Must-Be-Zero RMS Error: Making Room for New File Options in the FAB . . . . . 5–17
5.23 HP COBOL Run-Time Library (RTL) . . . . . 5–18
5.23.1 Performance Improvement of COBOL CALL Statement . . . . . 5–18
5.24 HP Fortran for Integrity servers . . . . . 5–18
5.25 HP MACRO for OpenVMS . . . . . 5–19
5.25.1 Enhancements for the Macro-32 Compiler . . . . . 5–19
5.25.2 HP MACRO for OpenVMS Integrity servers . . . . . 5–21
5.25.3 HP MACRO for OpenVMS Alpha Systems . . . . . 5–22
5.25.4 /OPTIMIZE=VAXREGS Qualifier Not Supported on Integrity servers . . . . . 5–22
5.25.5 Floating Divide-by-Zero Error Not Raised (Integrity servers Only) . . . 5–22
5.26 Hypersort Utility . . . . . 5–23
5.26.1 Reporting a Problem to HP . . . . . 5–23
5.26.2 Large Files Restriction . . . . . 5–23
5.26.3 Hypersort and VFC Files Restriction . . . . . 5–23
5.26.4 /FORMAT=RECORD_SIZE Restriction . . . . . 5–23
5.26.5 Using Hypersort with Search Lists and Other Uses of Logical Names . . . . . 5–24
5.26.6 Lack of Free Space for Work Files . . . . . 5–24
5.26.7 Input Asterisk (*) Restriction . . . . . 5–24
5.26.8 Optimal Working Set Extent and Page File Quota Settings . . . . . 5–24
5.27 Intel Assembler (Integrity servers Only) . . . . . 5–24
5.28 Librarian Utility . . . . . 5–24
5.28.1 Linking Against Data-Reduced ELF Object Libraries Not Recommended (Integrity servers Only) . . . . . 5–24
5.28.2 Failure to Insert or Replace .STB files in an Integrity servers Library (Integrity servers Only) . . . . . 5–25
5.28.3 Librarian Fails to Report Errors When Process Quota Too Low . . . . . 5–25
5.29 Linker Utility for OpenVMS Alpha . . . . . 5–25
5.29.1 Linker Appears to Hang When Many Files Are Specified . . . . . 5–26
5.29.2 Change in Linker Default Behavior with Library Check . . . . . 5–26
5.29.3 Limit of 25 Elements on Stack . . . . . 5–26
5.30 Linker Utility for OpenVMS Integrity servers . . . . . 5–27
5.30.1 Linker Writes Incorrect Interimage Debug Fixups into Debug Symbol File . . . . . 5–27
5.30.2 /SELECTIVE_SEARCH Qualifier Might Incorrectly Ignore Transfer Address . . . . . 5–27
5.30.3 Maximum Number of Sections . . . . . 5–28
5.30.4 Incorrect Creation Date of Shareable Images in the Map File . . . . . 5–28
5.30.5 Demangler Information Look Up Results in Linker Access Violation . . . . . 5–29
5.30.6 Incorrect Secondary Messages for the NOGLOSYM Error Message . . . 5–29
5.30.7 Incorrect Information for Undefined Symbols . . . . . 5–29
5.30.8 Incorrect UNMAPFIL Error . . . . . 5–29

5.30.9 Max Ident Length Change for Shareable Images in Map . . . . . 5–29
5.30.10 Linkage Type Check for Shareable Images . . . . . 5–29
5.30.11 Program Section Attribute ABS Ignored . . . . . 5–30
5.30.12 Linker ACCVIOs when FP_MODE Literal Missing From Command Line . . . . . 5–30
5.30.13 OpenVMS Integrity servers Object Module and Image File Information Currently Unavailable . . . . . 5–30
5.30.14 Differences Between the Integrity servers Linker and the Alpha Linker . . . . . 5–30
5.30.15 LINK_ORDER Section Header Flag Not Supported . . . . . 5–30
5.30.16 Linking Against Data-Reduced ELF Object Libraries Not Recommended . . . . . 5–31
5.30.17 Error in Handling Initialized Overlaid Program Sections Fixed . . . . . 5–31
5.30.18 Removal of Linker Qualifiers /EXPORT_SYMBOL_VECTOR and /PUBLISH_GLOBAL_SYMBOLS . . . . . 5–32
5.30.19 Support for Longer Symbol Names in Options . . . . . 5–32
5.30.20 Better Use of Memory for Linker-Created Code Stubs . . . . . 5–32
5.30.21 Compiler Support for Demangled Symbol Names . . . . . 5–32
5.31 LTDRIVER: CANCEL SELECTIVE Restriction . . . . . 5–32
5.32 Mail Utility: Threads Restriction for Callable Mail . . . . . 5–33
5.33 OpenVMS System Dump Analyzer (SDA) . . . . . 5–33
5.33.1 CLUE Commands Not Ported to OpenVMS Integrity servers . . . . . 5–33
5.34 PL/I Libraries Not Included in OpenVMS Integrity servers Version 8.2 . . . 5–33
5.35 POSIX Threads Library . . . . . 5–33
5.35.1 Support for Process-shared Objects . . . . . 5–34
5.35.2 New Return Status for pthread_mutex_lock . . . . . 5–34
5.35.3 Support for New API pthread_mutex_tryforcedlock_np . . . . . 5–34
5.35.4 Stack Overflows During Exception Handling (Integrity servers Only) . . . . . 5–35
5.35.5 THREADCP Command Behavior on Integrity Servers . . . . . 5–36
5.35.6 Floating-Point Compilations and Exceptions (Integrity servers Only) . . . . . 5–36
5.35.7 C Language Compilation Header File Changes . . . . . 5–36
5.35.8 New Priority Adjustment Algorithm . . . . . 5–37
5.35.9 Process Dumps . . . . . 5–37
5.35.10 Dynamic CPU Configuration Changes . . . . . 5–37
5.35.11 Debugger Metering Function Does Not Work . . . . . 5–38
5.36 RTL Library (LIB$) . . . . . 5–38
5.36.1 RTL Library (LIB$) Help Omission . . . . . 5–38
5.36.2 RTL Library (LIB$): Calling Standard Routines (Integrity servers Only) . . . . . 5–38
5.37 Screen Management (SMG$) Documentation . . . . . 5–39
5.38 SORT32 Utility . . . . . 5–40
5.38.1 CONVERT Problem With DFS-Served Disks . . . . . 5–40
5.38.2 Temporary Work Files Not Always Deleted . . . . . 5–40
5.38.3 SORT/SPECIFICATION With Compound Conditions: Requirement . . . . . 5–40
5.38.4 Performance Problem with Variable Length Records . . . . . 5–40
5.38.5 Work File Directories Restriction . . . . . 5–41
5.39 Timer Queue Entries (TQEs) . . . . . 5–41
5.40 Watchpoint Utility (Integrity servers Only) . . . . . 5–41
5.41 Whole Program Floating-Point Mode (Integrity servers Only) . . . . . 5–41

## 6 Hardware Release Notes


6.1 USB Device Support . . . . . 6–1
6.2 MP and BMC Console Restrictions (Integrity servers Only) . . . . . 6–2
6.2.1 Input, Output, and Error Device Restriction . . . . . 6–2
6.2.2 Remapping Ctrl/H to the Delete Key . . . . . 6–2
6.3 AlphaServer 2100 . . . . . 6–2
6.3.1 Console Display . . . . . 6–3
6.3.2 SCSI Controller Restriction . . . . . 6–3
6.4 AlphaServer 8200/8400: FRU Table Error . . . . . 6–4
6.5 AlphaServer ES47/ES80/GS1280 Systems . . . . . 6–4
6.5.1 INIT Console Command Usage on ES47/ES80/GS1280 Soft Partitions . . . . . 6–4
6.5.2 RAD Support . . . . . 6–4
6.5.3 License Requirements . . . . . 6–4
6.5.4 STOP/CPU and Shutdown Behavior . . . . . 6–4
6.5.5 Setting Time at MBM . . . . . 6–5
6.6 AlphaServer GS Series Systems . . . . . 6–5
6.6.1 AlphaServer GS80/160/320 Systems: Device Restriction . . . . . 6–5
6.6.2 OpenVMS Galaxy License Enforcement . . . . . 6–5
6.6.3 Installing Licenses . . . . . 6–5
6.6.4 AlphaServer GS60/GS60E/GS140 Multiple I/O Port Module Configuration Restriction . . . . . 6–7
6.7 AlphaStation 200/400: ISA_CONFIG.DAT Changes Required . . . . . 6–7
6.8 AlphaStation 255: PCI Configuration Restriction . . . . . 6–8
6.9 ATI RADEON 7000 Graphics (Integrity servers Only) . . . . . 6–8
6.9.1 Integrity servers Graphics Support . . . . . 6–8
6.9.2 Hardware Accelerated 3D Graphics Not Supported on RADEON 7000 . . . . . 6–9
6.10 ATI RADEON 7500 Graphics . . . . . 6–9
6.10.1 Resource Requirements . . . . . 6–9
6.10.2 DECW$OPENGLSHR_RADEON.EXE Renamed to DECW$MESA3DSHR_RADEON.EXE . . . . . 6–10
6.10.3 Support Limitations . . . . . 6–10
6.10.4 Video Artifacts at High Refresh Rates . . . . . 6–10
6.10.5 OpenGL Supports IEEE Arithmetic Only . . . . . 6–10
6.10.6 DECwindows Server Hangs When Output Is Written to the Graphics Console (OPA0) . . . . . 6–10
6.10.7 Monitor Must Be Connected During Initialization . . . . . 6–11
6.10.8 Boot Reset Recommendation (Alpha Only) . . . . . 6–11
6.10.9 No Overlay Planes . . . . . 6–11
6.10.10 Single Colormap . . . . . 6–11
6.10.11 Single Bit Depth for All Windows . . . . . 6–11
6.10.12 Pixel Depth for Read/Write Color Map . . . . . 6–12
6.10.13 Backing Store/Save Unders Not Supported for 3D Windows . . . . . 6–12
6.10.14 Threads Restriction . . . . . 6–12
6.10.15 No Support for Single Buffered Visuals . . . . . 6–12
6.10.16 No 3D Support for Color Index Mode . . . . . 6–12
6.10.17 Timer Mechanism . . . . . 6–12
6.11 DECwindows X11 Display Server (Alpha Only) . . . . . 6–13
6.11.1 S3 Multihead Graphics . . . . . 6–13
6.12 DIGITAL Modular Computing Components (DMCC) . . . . . 6–13
6.12.1 Alpha 5/366 and 5/433 PICMG SBC Restriction . . . . . 6–13
6.12.2 Updating the SRM Console . . . . . 6–13
6.13 Digital Personal Workstation: Booting OpenVMS V7.3-1 and Higher . . . . 6–14

6.14 Dual-Controller HSG*nn* with Many LUNs Can Fail Under Heavy I/O Load . . . . . 6–14
6.15 Open3D Graphics Licensing Change . . . . . 6–15
6.16 PowerStorm 300/350 PCI Graphics Support for OpenVMS . . . . . 6–15
6.17 RF*nn* DSSI Disk Devices and Controller Memory Errors . . . . . 6–16
6.18 RZ*nn* Disk Drive Considerations . . . . . 6–18
6.18.1 RZ25M and RZ26N Disk Drives: Recommendations . . . . . 6–18
6.18.2 RZ26N and RZ28M Disks: Recommended Firmware Support . . . . . 6–19
6.18.3 RZ26L and RZ28 Disks: Required Firmware for Multihost Use . . . . . 6–19
6.18.3.1 Firmware Revision Level 442 Requirements . . . . . 6–19
6.18.3.2 Firmware Revision Level 442 Installation Procedure . . . . . 6–19
6.19 sx1000 Integrity Superdome . . . . . 6–20
6.20 ZLX Graphics Boards Support . . . . . 6–20
6.21 Recompiling and Relinking OpenVMS Device Drivers . . . . . 6–20
6.21.1 Alpha and VAX SCSI Device Drivers . . . . . 6–20
6.21.2 OpenVMS Alpha Device Drivers . . . . . 6–21
6.22 Device Driver MON Version Handling . . . . . 6–21
6.23 Possible Per-Threads Security Impact on Alpha Device Drivers . . . . . 6–21
6.24 Device IPL Setup for OpenVMS Alpha Drivers . . . . . 6–21
6.25 CRCTX Routines Enhanced . . . . . 6–22
6.26 Adapter Release Notes . . . . . 6–22
6.26.1 Fibre Channel EFI Driver and Firmware Requirements . . . . . 6–22
6.26.2 Booting with Multiple Fibre Channel Boot Entries . . . . . 6–23

## A Interlocked Memory Instructions (Alpha Only)

A.1 Required Code Checks . . . . . A–1
A.2 Using the Code Analysis Tool (SRM_CHECK) . . . . . A–1
A.3 Noncompliant Code Characteristics . . . . . A–2
A.4 Coding Requirements . . . . . A–3
A.5 Compiler Versions . . . . . A–5
A.6 Recompiling Code with ALONONPAGED_INLINE or LAL_REMOVE_FIRST . . . . . A–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 . . . . . 4–26
4–2 Patch Kits Required for Cluster Compatibility . . . . . 4–27
4–3 Galaxy Definitions . . . . . 4–30
4–4 TFF Character Fallback Tables . . . . . 4–33
5–1 Macro-32 New Built-ins . . . . . 5–19
6–1 Changes to Device Description Block . . . . . 6–8
6–2 Supported Microcode Revision Levels . . . . . 6–17
6–3 Commands for Updating Microcode in Certain DSSI Disk Devices . . . 6–18
6–4 Revision Level 442 Firmware Compatibility . . . . . 6–19
A–1 Versions of OpenVMS Compilers . . . . . A–5

# Preface

## Intended Audience

This manual is intended for all users of the HP OpenVMS for Integrity servers or HP OpenVMS Alpha Version 8.4 operating system. Read this manual before you install, upgrade, or use OpenVMS Version 8.4.

## Document Structure

This manual contains the following chapters and appendix:

- Chapter 1 contains release notes that pertain to upgrading or installing the OpenVMS Alpha operating system or installing the OpenVMS Integrity servers.
- 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 the 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 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 versions that still apply to the new release. A subheading for each release note indicates either the version of origin (for example, *V8.3*) or the version when an old note was last updated (for example, a note from Version 8.3 that was revised for Version 8.4 will be labeled with *V8.4*).

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, see the *HP OpenVMS Version 8.4 New Features and Documentation Overview* manual.

For additional information about HP OpenVMS products and services, see:

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

## Reader's Comments

HP welcomes your comments on this manual. Please send your comments or suggestions to:

`openvmsdoc@hp.com`

## How to Order Additional Documentation

For information about how to order additional documentation, see:

`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.)<br>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:<br>• Additional optional arguments in a statement have been omitted.<br>• The preceding item or items can be repeated one or more times.<br>• Additional parameters, values, or other information can be entered. |
| .<br>.<br>. | A vertical ellipsis indicates the omission of items from a code example or command format; the items are omitted because they are not important to the topic being discussed. |
| ( ) | In command format descriptions, parentheses indicate that you must enclose choices in parentheses if you specify more than one. |

| | |
|---|---|
| [ ] | In command format descriptions, brackets indicate optional choices. You can choose one or more items or no items. Do not type the brackets on the command line. However, you must include the brackets in the syntax for OpenVMS directory specifications and for a substring specification in an assignment statement. |
| \| | In command format descriptions, vertical bars separate choices within brackets or braces. Within brackets, the choices are optional; within braces, at least one choice is required. Do not type the vertical bars on the command line. |
| { } | In command format descriptions, braces indicate required choices; you must choose at least one of the items listed. Do not type the braces on the command line. |
| **bold type** | Bold type represents the introduction of a new term. It also represents the name of an argument, an attribute, or a reason. |
| *italic type* | Italic type indicates important information, complete titles of manuals, or variables. Variables include information that varies in system output (Internal error *number*), in command lines (/PRODUCER=*name*), and in command parameters in text (where *dd* represents the predefined code for the device type). |
| UPPERCASE TYPE | Uppercase type indicates a command, the name of a routine, the name of a file, or the abbreviation for a system privilege. |
| `Example` | This typeface indicates code examples, command examples, and interactive screen displays. In text, this type also identifies URLs, UNIX commands and pathnames, PC-based commands and folders, and certain elements of the C programming language. |
| - | A hyphen at the end of a command format description, command line, or code line indicates that the command or statement continues on the following line. |
| numbers | All numbers in text are assumed to be decimal unless otherwise noted. Nondecimal radixes—binary, octal, or hexadecimal—are explicitly indicated. |

# 1

# OpenVMS Software Installation and Upgrade Release Notes

This chapter contains information that you must know before installing or upgrading to OpenVMS Version 8.4. Topics of interest to both Alpha and Integrity server users are covered first. Later sections group notes of interest to users of specific platforms.

HP recommends that you read the following manuals before installing or upgrading OpenVMS Version 8.4:

- *HP OpenVMS Version 8.4 Release Notes* (this manual)
- *HP OpenVMS Version 8.4 New Features and Documentation Overview*
- *HP OpenVMS Version 8.4 Upgrade and Installation Manual*

For more information about associated products, see Chapter 2 and for hardware release notes, see Chapter 6.

## 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 Integrity servers, OpenVMS Alpha, and OpenVMS VAX is kept up to date at:

`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 8.3-1H1 to 8.4).

  **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 8.2 to 8.3).

  **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 8.2-1).

  **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 OpenVMS field test Version 8.4 will be made available in the field test website.

## 1.4 Networking Options

*V8.4*

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

- Either HP DECnet-Plus Version 8.4 for OpenVMS or HP DECnet Phase IV Version 8.4 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, and the ability to run DECnet over TCP/IP or OSI protocols.

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 website. You must use HP TCP/IP Services for OpenVMS Version 5.7 after upgrading to OpenVMS Version 8.4.

## 1.5 Disk Incompatibility with Older Versions of OpenVMS

*V8.3*

The OpenVMS Version 8.4 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 ensure that the disk is compatible for that operating system version. For detailed instructions, see the *HP OpenVMS Version 8.3 Upgrade and Installation Manual*.

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, 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

*V8.4*

The following table lists the versions of DECwindows Motif supported on various platforms of the OpenVMS Version 8.4 operating system.

**Table 1–1 Supported Versions of DECwindows Motif**

| OpenVMS Version | DECwindows Motif Version |
|---|---|
| OpenVMS Integrity servers Versions 8.3, 8.3-1H1, and 8.4 | DECwindows Motif for OpenVMS Integrity servers Version 1.7 |
| OpenVMS Alpha Version 8.4 | DECwindows Motif for OpenVMS Alpha Version 1.7 |

The DECwindows Motif software relies on specific versions of OpenVMS server and device driver images. Ensure you 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, see the *HP DECwindows Motif for OpenVMS Installation Guide*.

# 1.7 OpenVMS Integrity server Users

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

## 1.7.1 Storage Controllers

*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).

OpenVMS Integrity servers Version 8.3 supports 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 2 GB Fibre Channel Universal PCI-X adapter (A6826A). This adapter allows connectivity to any external SAN-based Fibre Channel storage infrastructure supported by OpenVMS.

Support for SAS based storage is provided from OpenVMS Integrity servers Version 8.3-1H1 onwards.

#### Restrictions

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

- The U160 adapter (A6829A) is not officially supported on OpenVMS Integrity servers Version 8.3 and later, 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 environment, requires you to move to 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 Integrity servers Version 8.3 and later, and customers using them must upgrade to the A6826A FC adapter before running production applications on Version 8.2.

### 1.7.2 U160 SCSI Support for rx7620 and rx8620

*V8.3*

The rx7620 and rx8620 systems have an internal U160 (SCSI), which is included in 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 (using cables) to a SCSI rack.

### 1.7.3 Clearing the System Event Log on Integrity servers

*V8.3*

HP Integrity servers maintain a System Event Log (SEL) within the system console storage. OpenVMS Integrity servers 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 continue when the BMC SEL is full by following the prompts; OpenVMS will process the contents of the SEL. 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, see the *HP OpenVMS Version 8.3 Upgrade and Installation Manual*.

## 1.7.4 Firmware for Integrity Servers

*V8.4*

OpenVMS Integrity servers Version 8.4 was tested with the latest firmware for each of the supported Integrity servers.

For the entry-class Integrity servers, HP recommends that use the most current system firmware. For information about updating the system firmware for entry-class Integrity servers, see the *HP OpenVMS Version 8.4 Upgrade and Installation Manual*. (For rx7620, rx8620, and Superdome servers, call HP Customer Support to update your firmware.)

Table 1–2 lists the recommended firmware versions for entry-class Integrity servers:

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

| System | System Firmware | BMC Firmware | MP Firmware | DHPC Firmware |
|---|---|---|---|---|
| rx1600 | 4.27 | 4.01 | E.03.30 | N/A |
| rx1620 | 4.27 | 4.01 | E.03.30 | N/A |
| rx2600 | 2.31 | 1.53 | E.03.32 | N/A |
| rx2620 | 4.29 | 4.04 | E.03.32 | N/A |
| rx4640 | 4.29 | 4.06 | E.03.32 | 1.10 |
| rx2660* | 4.11 | 5.24 | F.02.23 | N/A |
| rx3600* | 4.11 | 5.25 | F.02.23 | 1.23 |
| rx6600* | 4.11 | 5.25 | F.02.23 | 1.23 |

*If you have Intel Itanium 9100 processors on your rx2660, rx3600, or rx660, you need firmware that is at least one version greater than the ones listed here.

For cell-based servers, you must access the MP Command Menu and issue the sysrev command to list the MP firmware revision level. The sysrev command is available on all HP Integrity servers that have an MP. Note the EFI info fw command does not display the Management Processor (MP) firmware version on cell-based Integrity servers.

To check firmware version information on an entry-class Integrity server that does not have the MP, enter the info fw command at the EFI prompt. Note the following example:

```
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 upgradeable 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 how to access and use EFI, see the *HP OpenVMS Version 8.4 Upgrade and Installation Manual*. For more information, refer to the hardware documentation that is provided with your server.

For instructions on upgrading your firmware for your entry-class Integrity servers, see the *HP OpenVMS Version 8.4 Upgrade and Installation Manual*. To upgrade firmware for the rx7620, rx8620, or Superdome, contact HP Customer Support.

### 1.7.5 Booting from the Installation DVD

*V8.2*

On Integrity servers with the minimum amount of supported memory (512 MB), 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 Booting from USB or vMedia Devices

*V8.4*

The %SYSTEM-I-MOUNTVER messages and the Universal Serial Bus Configuration Manager message are new to OpenVMS Version 8.4 and are seen only when using USB or vMedia devices for booting the Integrity rx2660, rx3600, and rx6600 servers.

### 1.7.7 HP DECwindows Motif Release Notes

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

#### 1.7.7.1 Keyboard Support

*V8.2*

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

## 1.8 OpenVMS Alpha Users

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

### 1.8.1 Firmware for OpenVMS Alpha Version 8.4

*V8.4*

OpenVMS Alpha Version 8.4 was tested with the platform-specific firmware included on Alpha Systems Firmware CD Version 7.3. For older platforms no longer included on the Firmware CD, OpenVMS Alpha Version 8.4 was tested with the latest released firmware version. HP recommends upgrading to the latest firmware before upgrading OpenVMS.

Read the firmware release notes before installing the firmware. For Version 7.3 and the latest firmware information, see:

`http://h18002.www1.hp.com/alphaserver/firmware/`

### 1.8.2 Upgrade Paths

*V8.4*

**If you are running OpenVMS Alpha or Integrity servers Version E8.4, you can upgrade directly to Version F8.4. If you are running other versions of OpenVMS, you must first install Version E8.4 and you can upgrade to Version F8.4.**

You can upgrade *directly* to OpenVMS Alpha Version 8.4 from only the following versions of OpenVMS Alpha.

For Alpha:

Version 7.3-2 to V8.4
Version 8.2 to V8.4

Version 8.3 to V8.4

For Integrity servers:

Version 8.2-1 to V8.3
Version 8.3 to V8.4
Version 8.3-1H1 to V8.4

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 8.4. 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 8.4:

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 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 these versions require the installation of one or more remedial kits. For more information, see Section 4.36.1.

**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 Integrity servers Version 8.4. For more information, refer to the *HP OpenVMS Version 8.4 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 8.4, and for instructions on installing OpenVMS Integrity servers Version 8.4, refer to the *HP OpenVMS Version 8.4 Upgrade and Installation Manual*.

## 1.9 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
```

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

To verify the 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 in the example, 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*.

## 1.10 Modifying SYSTARTUP_VMS.COM

*V8.4*

The startup command procedures for Encrypt and SSL are now called from the VMS$LPBEGIN-050_STARTUP.COM procedure. If you are upgrading from a previous version of OpenVMS that had Encrypt and SSL products installed, edit SYS$MANAGER: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 Encryption for OpenVMS

*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 1.12.

## 1.12 Upgrading HP DECram V3.*n*

*V8.2*

Starting with OpenVMS Alpha and OpenVMS Integrity servers 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 8.4 Upgrade and Installation Manual* for details.

More DECram release notes are included in Section 2.17.

## 1.13 Converting the LANCP Device Database

*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
```

## 1.14 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 in AUTOGEN behavior 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.

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

## 1.15 Remove TIE Kit Before Upgrade

*V8.2-1*

The Translated Image Environment (TIE) has been integrated into OpenVMS Integrity servers 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 TIE PCSI kit (HP-I64VMS-TIE) on OpenVMS Integrity servers Version 8.2 or Version 8.2-1, you must manually remove the TIE kit before you upgrade to OpenVMS Integrity servers 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 Integrity servers Version 8.2-1 or later.

## 1.16 Installation Failure of Layered Products on Alternate Devices or Directories

*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 of 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 lists 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 Integrity servers.

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 is available at:

`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 HP TCP/IP Services for OpenVMS

*V8.4*

You must use HP TCP/IP Services for OpenVMS Version 5.7 after upgrading to OpenVMS Version 8.4.

## 2.3 NetBeans Version 5.5.1 Requires Latest JDK

*V8.4*

NetBeans Version 5.5.1 for OpenVMS Alpha and OpenVMS Integrity servers is supported only on Java Platform, Standard Edition, Development Kit (JDK) v 1.4.2-7 or higher.

**Note**

JDK Version 6.0-1 for Integrity servers is not supported.

## 2.4 Problem Accessing DFS Mounted Disk

*V8.4*

When accessing an ODS-5 disk over DFS, if the path specification from the client includes the MFD, the access fails with the following error message as shown in the example. If the path specification does not include the MFD, the access succeeds.

**Example:**

Let the access point be DKA100:[USERS] and the client DFS disk be DFSC1001.

```
$ DIR DFSC1001:[000000]  ! fails with the following error message:

%DIRECT-E-OPENIN, error opening DFSC1001:[000000]*.*;*
as input -RMS-E-DNF, directory not found
-SYSTEM-W-NOSUCHFILE, no such file

$ TYPE DFSC1001:[000000]file.dat  ! fails with the following error message:
where:
file.dat is a file under USERS.DIR.

%TYPE-W-SEARCHFAIL, error searching for DFSC1001:[000000]file.dat
-RMS-E-DNF, directory not found
-SYSTEM-W-NOSUCHFILE, no such file

$ DIR DFSC1001:[SUBDIR]  ! works as expected
where:
SUBDIR is a subdirectory under USERS.DIR.
```

## 2.5 HP DCE for OpenVMS Restriction (Integrity servers Only)

*V8.4*

On OpenVMS Version 8.4, if you install HP DCE for OpenVMS Version 3.2 by selecting **Install or Upgrade Layered Products** from the **Install** menu, it fails with the following error message:

```
%PCSI-E-ERROWNER, error in owner specification 'DCE$SERVER'
-SYSTEM-E-NORIGHTSDB, rights database file not found %PCSI-E-OPFAILED,
operation failed
```

#### Workaround

To install DCE, mount the OE DVD installation disk on the OpenVMS system and execute the DCE$INSTALL.COM procedure which is present under DCE_I64032 folder.

## 2.6 XML-C Product Zip File

*V8.3-1H1*

The XML-C product for OpenVMS for Integrity servers is delivered as a ZIP file that contains a self-extracting executable file. The XML-C installation documentation describes how to install the product by using this executable file. To obtain the executable file, extract it from the ZIP file.

## 2.7 OpenVMS e-Business and Integration Infrastructure Package

*V8.3-1H1*

The OpenVMS e-Business and Integration Infrastructure Package for OpenVMS is contained on two CDs that are formatted so that they appear as a Files-11 file structure to an OpenVMS system and an ISO 9660 file structure to a Windows, Linux, or UNIX system.

**Installation**

The component installation kits and documentation are split across the two CDs. Component installation can be done only on an OpenVMS Alpha system from the specific CD designated in the top-level `index.html` file.

**Documentation**

For OpenVMS systems, partial component documentation is viewable based on which CD is mounted for use. Component documentation is available only for the components present on the specific CD.

For Windows, Linux, or UNIX systems, complete component documentation is viewable on both CDs.

## 2.8 Updates to Freeware Readme File

*V8.3-1H1*

An update to the [FREEWARE]FREEWARE_README.TXT, included on each OpenVMS Freeware CD, is available for download from the OpenVMS Freeware website at:

`http://h71000.www7.hp.com/openvms/freeware/index.html`

This updated file includes the correction to the displayed version number from V7.0 to the intended and expected V8.0, as well as additional updates and corrections.

As is traditional with the OpenVMS Freeware, all updates to existing files and new packages are available at the OpenVMS Freeware website.

## 2.9 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

## 2.10 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.

## 2.11 COM for HP OpenVMS (Alpha Only)

The following release notes pertain to HP COM for OpenVMS.

### 2.11.1 COM for OpenVMS Support

*V8.2*

HP 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/`

### 2.11.2 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.

## 2.12 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.

## 2.13 DECforms Web Connector Version 3.0 (Alpha Only)

*V7.3-1*

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

1. Remove or comment out the following line:

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

   from these command procedures in the FORMS$INSTALL_AREA directory:

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

2. Ensure that the Java environment is set up systemwide for *all* processes. HP recommends adding the Java environment setup to the system's SYSLOGIN.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.

## 2.14 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 effected 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.

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 5.34.

## 2.15 FMS Kits

*V8.3*

You can install either of the following FMS kits (or later versions) on both OpenVMS Alpha and OpenVMS Integrity servers:

Full kit: HPFMS025

Run-time kit: HPFMSRT025

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

## 2.16 Graphical Configuration Manager (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/
```

## 2.17 HP DECram

This section contains release notes pertaining to DECram.

**Note**

Refer to Section 1.12 for more information on HP DECram.

### 2.17.1 DECram Available with OpenVMS Version 8.2 and later

*V8.2*

Starting with OpenVMS Alpha and Integrity servers 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.
```

### 2.17.2 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"
```

## 2.18 HP DECwindows Motif for OpenVMS

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

### 2.18.1 New Locales Added

*V8.2*

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

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

### 2.18.2 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 later.

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.

## 2.19 HP Secure Web Server Version Support

*V8.2*

OpenVMS Alpha Version 7.3-2 and OpenVMS Version 8.2 (Alpha and Integrity servers) 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.

## 2.20 HP Pascal for OpenVMS Alpha Systems

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

### 2.20.1 HP Pascal: Version 5.8A (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 or later. The Pascal V5.8A kit contains an enhanced installation procedure to correctly build the STARLET library files on OpenVMS Version 7.3-2 or later.

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

### 2.20.2 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.

## 2.21 WEBES and SEA Support on Integrity servers

*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/`

# 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 8.4 New Features and Documentation Overview*.

## 3.1 SYS$GETTIM_PREC System Service Declaration

*V8.4*

The SYS$GETTIM_PREC system service declaration is not present in the FORTRAN library FORSYSDEF.TLB. Unlike other languages, FORSYSDEF.TLB ships with the FORTRAN compiler and is built against the lowest OS supported. In a future release, we will be providing OpenVMS V8.4 based FORSYSDEF library which contains the SYS$GETTIM_PREC system service declarartion along with the usual library built against the lowest OS.

## 3.2 Problem With F$GETSYI("RAD_CPUS")

*V8.4*

On a cell-based Integrity server system containing 64 CPUs, with both cell local memory (CLM) and interleaved memory (ILM) configured, the output of F$GETSYI("RAD_CPUS") does not include the 64th CPU as a member of the base RAD.

## 3.3 HP Code Signing Service for OpenVMS Support

*V8.4*

HP Code Signing Service (HPCSS) for OpenVMS supports PCSI and VMSINSTAL based kits.

## 3.4 Symbolic Links Implementation Changes

*V8.4*

Symbolic links (Symlinks) was first introduced with OpenVMS Version 8.3. The internal implementation of Symlinks has been improved in OpenVMS Version 8.4.

### 3.4.1 Logical Names

*V8.4*

The new symlinks implementation allows the use of logical names as the first element of a target pathname.

For example,

```
$ CREATE /SYMLINK="/SYS$HELP/CC_RELEASE_NOTES.PS" RELNOTES.PS
$ DIR /SIZE /NOSYMLINK RELNOTES.PS

Directory SYS$SYSROOT:[SYSMGR]

RELNOTES.PS;1            209

Total of 1 file, 209 blocks.
```

### 3.4.2 Audit Alarms Fixed

*V8.4*

With the previous implementation of symlinks, when commands such as $DIR attempted to list a directory, it resulted in audit alarms if the user did not have permissions on the target of a given file. DIRECTORY command must read the file header (that is, perform a file access operation) in order to determine if the file was a symlink. This access will trigger the audit alarm if the user issuing DIRECTORY command does not have read permissions on the target file. This problem has been corrected with the new symlink design on OpenVMS Version 8.4, where a directory entry indicating a symlink is flagged as such, with DIR$V_SPECIAL set to 1 (overlaid with DIR$V_NEXTREC).

**Note**

- Compatibility with symbolic links created by OpenVMS Version 8.3

  Symlinks created by OpenVMS Version 8.4 will work on OpenVMS Version 8.3. However, symlinks created by OpenVMS Version 8.3 will not work on OpenVMS Version 8.4.

  To convert symlinks created by OpenVMS Version 8.3 to the format required on OpenVMS Version 8.4, you must run the ANALYZE/DISK_STRUCTURE (VERIFY) utility with the /REPAIR qualifier.

## 3.5 SHOW SYSTEM/STATE=MUTEX Does not Display the Processes

*V8.4*

`$ SHOW SYSTEM/STATE=MUTEX` command does not display the processes in MUTEX state.

However, you can display the processes in MUTEX state by entering the following command:

```
$ SHOW SYSTEM
```

## 3.6 SWB V1.1-12 Installation Warnings

*V8.4*

SeaMonkey Version 1.0 is built based on the Mozilla Version 1.8b1 code. When you install SWB Version 1.1-12 on an OpenVMS server that already has SWB Version 1.7-13 installed, you will see the following warning message:

```
%PCSI-W-VERLOW, you have selected a lower version of an
installed product
-PCSI-W-VERINS, the installation of product HP I64VMS CSWB V1.1-12
-PCSI-W-VERREM, will remove current product HP I64VMS CSWB V1.7-13
         Do you want to continue? [YES]
```

This is because PCSI always considers a higher number for a new version and in this case the latest SWB's version number is lower than it's predecessor. You can ignore this warning.

## 3.7 Ctrl/P at the Console Does not Always Work

*Permanent Restriction*

On certain Integrity server configurations, pressing Ctrl/P at the console does not cause OpenVMS to display the Interrupt Priority C (IPC) menu. If you plan to use Ctrl/P, you should test it to ensure that it works.

If necessary, you can restore Ctrl/P functionality by performing the following steps:

1. Invoke SDA to analyze the running system:

   ```
   $ ANALYZE/SYSTEM
   ```

2. Use the CLUE CONFIG command to display the adapters on the system:

   ```
   SDA> CLUE CONFIG/ADAPTER
   ```

3. Locate the "Console Serial Line Driver" adapter (SRA:) in the display:

```
System Adapter Configuration:
-----------------------------
TR Adapter     ADP               Hose Bus    BusArrayEntry    Node GSIN  iVec  SCB  Port Slot Device Name/HW-Id
-- ----------- ----------------- ---- ---------------------- ---- ---------------- ---- ---- ----------------------
...
 5 ACPI_IA64_I FFFFFFFF.8832E0C0   0  00 IA64_BUS
 6 PCI         FFFFFFFF.88342A80   9  00 PCI
                                          FFFFFFFF.88342E58  0   0018  00DF 15F0  SRA: 0 Console Serial Line Driver
                                          FFFFFFFF.88342F68  8   0018  00DF 15F0  EWA: 1 A6865A (Fast Ethernet)
...
```

4. Identify the controller that shares the same Global System Interrupt Number (GSIN) as SRA:. In this example, it is EWA:.

5. Exit from SDA and enter the following command (substituting EWA with the correct controller):

   ```
   $ SET DEVICE EWA0/PREFERRED_CPUS='F$GETSYI("PRIMARY_CPUID")'
   ```

When you complete these steps, Ctrl/P should now function correctly. HP recommends that you edit SYS$MANAGER:SYLOGICALS.COM to include the SET DEVICE command to ensure correct behavior when the system reboots. Note that Ctrl/P might stop working again if you add or remove I/O adapters. If this happens, redo the steps listed above.

Also, note that if XDELTA or the System Code Debugger has been loaded when the system was booted, Ctrl/P is not affected. Entering Ctrl/P will cause the XDELTA prompt to be displayed, for example:

```
Console Brk at 807CF3D2 on CPU 0
807CF3D2!            cmp4.lt     p0, p6 = 3F, r4  (New IPL = 3)
```

## 3.8 Installing Oracle Database 10g Release 2 for OpenVMS Fails

*V8.4*

Due to intermittent problems during the installation of Oracle Database 10g Release 2 for OpenVMS on field test Version 8.4, the Oracle 10g Release 2 installation fails. HP is analyzing and will fix this problem in a future release.

If you are already using Oracle Database 10g Release 2 on prior versions of OpenVMS, you can upgrade the OpenVMS operating system to Version 8.4 and continue to use Oracle on OpenVMS Version 8.4.

## 3.9 Serial Port Enumeration

*V8.3-1H1*

On OpenVMS systems, enumeration is the process by which devices are assigned a letter and number following the OpenVMS generic device-type nomenclature. In the case of serial ports, enumeration is expressed as TTA0, TTB0, and so on, for generic serial port devices, and as OPA0 for a serial port device that has been selected as the system's primary console at the EFI Boot Manager or the EFI Shell> prompt.

OpenVMS Version 8.2 consistently enumerated system serial ports according to the rules and precedents established by OpenVMS Alpha systems. With OpenVMS Version 8.3, those rules were violated and users experienced inconsistent port naming, particularly on systems migrating from Version 8.2 to Version 8.3.

OpenVMS Version 8.3-1H1 returns to the consistent serial-port naming conventions established for HP Integrity in OpenVMS Version 8.2, with the goal of not changing serial port names more than necessary, and for consistency with the policy on OpenVMS Alpha systems. The names of the serial ports can change, because at least one serial port can serve more than one function.

The serial port selected as primary console is always OPA0. If the graphics console has been selected as primary, the keyboard and graphics head constitute OPA0, and the serial ports will be named TTA0, TTB0, and so on.

Unless the serial port of the Integrated Lights Out (iLO) Management Processer (MP) is selected as the primary console, it is not connected as a serial port and is not exposed by the operating system. It is not suitable for general-purpose use because it cannot support the data rates a general-purpose serial port needs to support. This is an optional component in most systems. Check the options list shipped with your system and your system's documentation at the HP documentation website:

`http://docs.hp.com`

There are two possible serial ports that can be selected as the primary console, the iLO and the Baseboard Management Console (BMC). Whichever is selected as primary console will be expressed as OPA0 by OpenVMS; the other will be either TTA0 or TTB0 if the system has an additional serial port. The following list describes these abbreviations and their definitions.

| Abbreviation | Definition |
|---|---|
| MP | Serial port of the iLO MP. This component is optional on some systems. |
| BMC | Serial port of the BMC. This component is not available on all systems. |
| AP | Auxiliary Port. This auxiliary 16550-compatible serial port is not available on all systems. |
| VGA | Graphics Console. This is an optional component of the iLO MP. If your system was not shipped with the VGA option, you can install a graphics option in one of the PCI slots to obtain this functionality. |
| NA | Not Available. |
| NC | Not Configured as a Serial Port by OpenVMS. |
| NS | Not Supported. |

The following table displays the sources for backpanel drawings:

| Platform | Backpanel Drawing |
|---|---|
| rx1600, rx1620 | http://docs.hp.com/en/AB430-96004/ch03s03.html#i1021437 |
| rx2600, rx2620 | http://docs.hp.com/en/AD117-9003A/ch02s02.html |
| rx4640 | http://docs.hp.com/en/A9950-96009/A9950-96009.pdf |
| rx3600, rx6600 | http://docs.hp.com/en/AD217-9001A/ch02s03.html |
| rx2660 | http://docs.hp.com/en/AD217-9001A/ch02s02.html |
| rx8620 | http://docs.hp.com/en/A7026-96033/ch04s05.html |
| bl860c | http://docs.hp.com/en/AD217-9001A/ch02s01.html |

The following table provides serial-port naming for the HP Integrity platforms listed. The device selected as primary console is always named OPA0.

| | | OpenVMS Serial Port Name | | | |
|---|---|---|---|---|---|
| Platform | Primary Console Port | MP | BMC | AP | VGA |
| rx1600<br>rx1620 | MP (optional)<br>BMC<br>VGA (optional) | OPA0<br>NC<br>NC | TTA0<br>OPA0<br>TTA0 | NA | OPA0 |
| rx2600<br>rx2620 | MP (optional)<br>BMC<br>VGA (optional) | OPA0<br>NC<br>NC | TTB0<br>OPA0<br>TTB0 | TTA0<br>TTA0<br>TTA0 | OPA0 |
| rx4640 | MP<br>VGA (optional) | OPA0<br>NC | NA | NA | OPA0 |

| | | OpenVMS Serial Port Name | | | |
|---|---|---|---|---|---|
| **Platform** | **Primary Console Port** | **MP** | **BMC** | **AP** | **VGA** |
| rx3600<br>rx6600 | MP (optional)<br>BMC<br>VGA (optional) | OPA0<br>NC<br>NC | TTA0<br>OPA0<br>TTA0 | OPA0 | |
| rx2660 | MP<br>VGA (optional) | OPA0<br>NC | NA | TTA0<br>TTA0 | OPA0 |
| rx8620 | MP<br>VGA | OPA0<br>NC | NA | NA | OPA0 |
| BL860c | MP<br>VGA | OPA0<br>NC | NA | NA | OPA0 |

## 3.10 Old Firmware Cannot Translate Messages Written to the System Event Log

*8.3-1H1*

Upon installation, OpenVMS Version 8.3-1H1 begins writing new messages to the system event log. To see the SEL, select it (on most systems) from the main MP menu (SL: Show Event Logs).

Older firmware translates the messages as "IPMI Type-E0 Event," instead of the correct OS_OPENVMS_BUGCHECK and OS_OPENVMS_SHUTDOWN.

The following shows the OS_OPENVMS_BUGCHECK message (alert level *5 - critical) on a system running older firmware:

```
291        0  *5  0xB4801C9700E01B50 000000000019000C IPMI Type-E0 Event
                                                     30 Jul 2007 14:03:41
```

The following is an example of OS_OPENVMS_SHUTDOWN message (alert level 2 - informational) on a system running older firmware:

```
296        0   2  0x54801C9900E01BD0 00000000001A000C IPMI Type-E0 Event
                                                     30 Jul 2007 14:22:06
```

The new firmware uses the phrase "OS_OPENVMS_BUGCHECK" or "OS_ OPENVMS_SHUTDOWN" in place of "IPMI Type-E0 Event".

A third message, OS_BOOT_COMPLETE, has a different alert level on a system running new firmware. It has been changed by OpenVMS to informational, or level 2:

```
301   OS   0   2  0x548016E100E01B80 0000000000000001 OS_BOOT_COMPLETE
                                                     23 Aug 2007 14:25:44
```

New firmware displays the following message when "T - View Mode Configuration Text" is selected:

```
MP:SL (+,-,CR,D, F, L, J, H, K, T, A, U, ? for Help, Q or Ctrl-B to Quit) >t .
.
.
Log Entry 301: 23 Aug 2007 14:25:44
Alert Level 2: Informational
Keyword: OS_BOOT_COMPLETE
OS Boot Complete
Logged by: O/S Kernel (Generic)  0
Data: Major change in system state - Boot Complete 0x548016E100E01B80 0000000000000001
```

## 3.11 TZ Function in C RTL

*V8.3-1H1*

The TZ logical name or DCL symbol is used by the C Run-Time Library (C RTL) to define the time zone to be used in certain C program time-related functions. (For more information about TZ, its use, and specific functions, see the C Run-Time Library documentation.)

The TZ logical name or DCL symbol has been used by the C Run-Time Library since Version 7.3 of OpenVMS. However, with Version 8.3, there has been a change.

Prior to Version 8.3, defining TZ to something other than a valid time zone caused the time zone to default to local time (that is, the current time zone of your system). With OpenVMS Version 8.3, defining TZ to an invalid time zone causes the C RTL functions to resort to Coordinated Universal Time (UTC) time.

Note that if you define the logical name or DCL symbol TZ to a non-standard definition, it might cause undesirable side-effects in some C programs.

## 3.12 InfoServer Utility and FDDI

*V8.3-1H1*

Using the InfoServer utility on OpenVMS to boot a client over an FDDI network adapter is not supported.

## 3.13 New Qualifier for DCL Command SET PASSWORD

*V8.3-1H1*

The DCL command SET PASSWORD now accepts the /PROMPT qualifier with two permitted values: /PROMPT=FIXED and /PROMPT=VARIABLE. If you use the SET PASSWORD command in a DCL command procedure, do not specify the /PROMPT=VARIABLE qualifier. If you do, it works as expected, but any failing status is only displayed and not returned to DCL.

## 3.14 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.

After 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.

### 3.14.1 Freeware Menu Unavailable (Integrity servers Only)

*V8.2*

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

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 Integrity servers, 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.

## 3.15 DCL Commands

The following release notes pertain to DCL commands.

### 3.15.1 SHUTDOWN.COM on OpenVMS Graphics Console (Integrity servers only)

*Permanent Restriction*

On an OpenVMS Integrity server system with a graphics console, use of SYS$SYSTEM:SHUTDOWN.COM to stop the system may not work as expected. The system will not stop after the "SYSTEM SHUTDOWN COMPLETE" message and wait for a key to be typed at the console keyboard. However, it will continue to reset as though a reboot had been requested. If the first option in the list of boot options is a valid boot device, the OpenVMS system will reboot.

### 3.15.2 DIAGNOSE Command No Longer Supported

*V8.2*

The DIAGNOSE command is not supported on OpenVMS Version 8.2.

### 3.15.3 MOUNT Command Restriction

*V8.4*

In a mixed-version OpenVMS cluster, an attempt to mount a volume with /CLUSTER and /CACHE=[NO]DATA from a OpenVMS Version 8.4 system will fail on the earlier versions of OpenVMS (%MOUNT-W-RMTMNTFAIL) with the error condition as MOUNT-F-BADPARAM.

For more information about enabling or disabling XFC while mounting a volume, see the *HP OpenVMS Version 8.4 New Features and Documentation Overview*.

## 3.16 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. 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`

## 3.17 HP Secure Web Browser

The following notes pertain to the HP Secure Web Browser.

### 3.17.1 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.

### 3.17.2 Installation Error on ODS-2 Disk Volume (Integrity servers Only)

*V8.2*

Installing the HP Secure Web Browser (CSWB) Version 1.4 for OpenVMS Integrity servers 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.

# 3.18 Documentation Corrections

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

## 3.18.1 *HP OpenVMS Linker Utility Manual* Update

*V8.4*

### 3.18.1.1 HP C++ Examples

In section 2.6.2, the following should be appended to point 7:

Note that on Integrity servers, you can use either the CXXLINK command or invoke the OpenVMS Linker to combine your object modules into one executable image. On OpenVMS Alpha, you must use the CXXLINK utility to link the object modules into one executable image. On Integrity server systems, the only benefit of using CXXLINK is that CXXLINK reports non-mangled names of undefined multiply-defined. It does this by intercepting Linker diagnostics and converting mangled names reported by the Linker to their original names, using the information in the demangler database.

## 3.18.2 *HP PCSI Utility Online help and Manual*: $PRODUCT REGISTER VOLUME Syntax Error Correction

*V8.4*

The HP PCSI utility online help defines incorrect syntax of the $PRODUCT REGISTER VOLUME command as follows:

```
$PRODUCT REGISTER VOLUME old-volume-label device-name
```

The correct syntax is as follows:

```
$PRODUCT REGISTER VOLUME old-logvolnam device-name
```

## 3.18.3 iCAP Release Notes: GiCAP Functionality not Available

*V8.3-1H1*

While running SYS$MANAGER:ICAP$CONFIG.COM, if you respond "Y" to the "Enter (Y)es to configure this system with GiCAP support (N):" prompt, the following message is displayed:

```
               HP OpenVMS Industry Standard 64
Global Instant Capacity on Demand (GiCAP) configuration utility
    *** GiCAP functionality is not currently available ***
  ***GiCAP will be enabled at a later date via an ECO kit ***


HP OpenVMS Industry Standard 64
Global Instant Capacity on Demand (GiCAP) configuration utility
*** GiCAP functionality is not currently available ***
 ***GiCAP will be enabled at a later date via an ECO kit ***
```

Also, note that in the release notes for Instant Capacity (iCAP), Chapter 2 specifies GiCAP support for OpenVMS Version 8.3-1H1. This support is not available currently, but will be available in a future update kit. For more information, see the OpenVMS website.

## 3.18.4 *POLYCENTER Software Installation Utility Developer's Guide*: PRODUCT Command Update

*V8.4*

In the parameters section, the *producer* description should be as follows:

Indicates the legal owner of the software product. This parameter must be either a double quoted or an unquoted string.

## 3.18.5 *HP OpenVMS System Manager's Manual Update*

*V8.4*

The following corrections pertain to the *HP OpenVMS System Manager's Manual, Volume 1: Essentials*.

### 3.18.5.1 Getting Information About Devices on the System

In section 8.3, the following examples should be replaced as follows:

The following command requests a full listing of the status of the DAD42: RRD40 device. The device is located on node IRIS in an OpenVMS Cluster environment.

```
$ SHOW DEVICES/FULL DAD42:

Disk DAD42: (IRIS), device type RRD40, is online, mounted, software write-locked,
file-oriented device, shareable, error logging is enabled.

 Error count                    0   Operations completed              146
 Owner process                 ""   Owner UIC                     [SYSTEM]
 Owner process ID       00000000   Dev Prot  S:RWPL,O:RWPL,G:RWPL,W:RWPL
 Reference count               1   Default buffer size               512
 Total blocks            1218000   Sectors per track                   4
 Total cylinders           50750   Tracks per cylinder                 6
 Allocation class             11

 Volume label     "CDBIN06JUL21"   Relative volume number              0
 Cluster size                  3   Transaction count                   1
 Free blocks               15153   Maximum files allowed          152083
 Extend quantity               5   Mount count                         1
 Mount status             System   Cache name       "_$11$DUA21:XQPCACHE"
 Extent cache size            64   Maximum blocks in extent cache   1515
 File ID cache size           64   Blocks currently in extent cache    0
 Quota cache size              0   Maximum buffers in FCP cache     1330
Volume status: ODS-2, subject to mount verification, file high-water marking,
 write-through XQP caching enabled, write-through XFC caching enabled.
```

The following command requests a full informational display about each DG device. This display shows only the first two devices: the mounted $1$DGA5001: device and the unmounted $1$DGA5004: device.

```
$ SHOW DEVICES/FULL DG

Disk $1$DGA5001: (CEAGLE), device type HSV110, is online, mounted,
file-oriented device, shareable, device has multiple I/O paths,
served to cluster via MSCP Server, error logging is enabled.

 Error count                    0    Operations completed             5773
 Owner process                 ""    Owner UIC                     [SYSTEM]
 Owner process ID        00000000    Dev Prot           S:RWPL,O:RWPL,G:R,W
 Reference count                1    Default buffer size               512
 Current preferred CPU Id       0    Fastpath                            1
 WWID 01000010:6005-08B4-0001-42DC-0001-F000-0111-0000
 Total blocks            20971520    Sectors per track                 128
 Total cylinders             1280    Tracks per cylinder               128
 Host name               "CEAGLE"    Host type, avail AlphaServer ES40, yes
 Alternate host name      "CLETA"    Alt. type, avail AlphaServer ES40, yes
 Allocation class               1

 Volume label              "5001"    Relative volume number              0
 Cluster size                  21    Transaction count                   1
 Free blocks             19598208    Maximum files allowed          476625
 Extend quantity                5    Mount count                         9
 Mount status              System    Cache name      "_$1$DGA3105:XQPCACHE"
 Extent cache size             64    Maximum blocks in extent cache 1959820
 File ID cache size            64    Blocks in extent cache              0
 Quota cache size               0    Maximum buffers in FCP cache     3444
 Volume owner UIC        [SYSTEM]    Vol Prot   S:RWCD,O:RWCD,G:RWCD,W:RWCD

Volume Status: ODS-2, subject to mount verification, file high-water marking,
 write-back XQP caching enabled, write-through XFC caching enabled.
Volume is also mounted on VMSROC, PAVER, VMSROL, CLETA, VMSJO, VMSMO, NOME,
 FARKLE.

I/O paths to device             5
Path PGA0.5000-1FE1-0015-22AC (CEAGLE), primary path.
  Error count                   0      Operations completed              0
Path PGA0.5000-1FE1-0015-22A9 (CEAGLE).
  Error count                   0      Operations completed              0
Path PGB0.5000-1FE1-0015-22A8 (CEAGLE).
  Error count                   0      Operations completed              0
Path PGB0.5000-1FE1-0015-22AD (CEAGLE), current path.
  Error count                   0      Operations completed           5773
Path MSCP (CLETA).
  Error count                   0      Operations completed              0

Disk $1$DGA5004: (CEAGLE), device type HSV110, is online,
file-oriented device, shareable, device has multiple I/O paths,
served to cluster via MSCP Server, error logging is enabled.

 Error count                    0      Operations completed              0
 Owner                    process      Owner UIC                   [SYSTEM]
 Owner process ID        00000000      Dev Prot         S:RWPL,O:RWPL,G:R,W
 Reference count                0      Default buffer size             512
 Current preferred CPU Id       0      Fastpath                          1
 WWID 01000010:6005-08B4-0001-42DC-0001-F000-0120-0000
 Host name                 CEAGLE    Host type, avail AlphaServer ES40, yes
 Alternate host name        CLETA    Alt. type, avail AlphaServer ES40, yes
 Allocation class               1
```

```
I/O paths to device 5
Path PGA0.5000-1FE1-0015-22AC (CEAGLE), primary path.
Error count                    0      Operations completed             0
Path PGA0.5000-1FE1-0015-22A9 (CEAGLE).
Error count                    0      Operations completed             0
Path PGB0.5000-1FE1-0015-22A8 (CEAGLE), current path.
Error count                    0      Operations completed             0
Path PGB0.5000-1FE1-0015-22AD (CEAGLE).
Error count                    0      Operations completed             0
Path MSCP (CLETA).
Error count                    0      Operations completed             0
```

### 3.18.5.2 Initializing a New Volume with ODS-5 Format

In Section 9.3.3, the SHOW/DEVICE DKA200:/FULL command displays the messages similar to the following:

```
$ SHOW DEVICE DKA200:/FULL

Disk $10$DKA200:, device type RZ74, is online, allocated, deallocate
on dismount, mounted, file-oriented device, shareable.

 Error count     0 Operations completed 155
 .
 .
 .
Volume Status: ODS-5, subject to mount verification, file high-water
marking, write-back XQP caching enabled, write-through XFC caching enabled.
```

### 3.18.5.3 Converting from ODS-2 to ODS-5

In section 9.5.5.1, the SHOW DEVICE DKA200:/FULL command in the instruction 2 should display the message similar to the following:

```
$ SHOW DEVICE DKA200:/FULL

Disk $10$DKA200:, device type RZ47, is online, allocated, deallocate
on dismount, mounted, file-oriented device, shareable.

 Error count     0 Operations completed 232
 .
 .
 .
Volume Status: ODS-2, subject to mount verification, file high-water
marking, write-back XQP caching enabled, write-through XFC caching enabled.
```

In section 9.5.5.1, the SHOW DEVICE DKA300:/FULL command in the instruction 5 displays the message similar to the following:

```
$ SHOW DEVICE DKA300:/FULL

Disk $10$DKA300:, device type RX74, is online, allocated, deallocate
on dismount, mounted, file-oriented device, shareable.

 Error count     0 Operations completed 155
 .
 .
 .
Volume Status: ODS-5, subject to mount verification, file high-water
marking, write-back XQP caching enabled, write-through XFC caching enabled.
```

### 3.18.5.4 New Extended File Specifications Characteristics

In section 10.1.2.1, the SHOW DEVICE command in the "Be Aware of Volume Structure" notes displays the message similar to the following:

```
$ SHOW DEVICE DKA500:/FULL

Disk AABOUT$DKA500:, device type DZ25 Disk, is online, allocated, deallocate
on dismount, mounted, file-oriented device, shareable.

 Error count     0 Operations completed 155
 .
 .
 .
Volume Status: ODS-5, subject to mount verification, file high-water
marking, write-back XQP caching enabled, write-through XFC caching enabled.



$ SHOW DEVICE DKA200:/FULL

Disk AABOUT$DSA200:, device type RZ25 Disk, is online, allocated, deallocate
on dismount, mounted, file-oriented device, shareable.

 Error count     0 Operations completed 232
 .
 .
 .
Volume Status: ODS-2, subject to mount verification, file high-water
marking, write-back XQP caching enabled, write-through XFC caching enabled.
```

### 3.18.5.5 ODS-2 and ODS-5 Used Together

In section 10.1.2.2, the SHOW DEVICE command example in the "Error Messages Can Vary Depending on Parse Style" notes should display the message similar to the following:

Examples of TRADITIONAL and EXTENDED styles on an ODS-5 volume:

```
$ SHOW DEVICE DKA500:/FULL

Disk AABOUT$DKA500:, device type RZ25 Disk, is online, allocated, deallocate
on dismount, mounted, file-oriented device, shareable.

 Error count     0 Operations completed 155
 .
 .
 .
Volume Status: ODS-5, [1] subject to mount verification, file high-water
marking, write-back XQP caching enabled, write-through XFC caching enabled.

$ SET PROCESS /PARSE_STYLE=TRADITIONAL [2]
$ OPEN /WRITE FILE z.z.z.z
%DCL-W-PARMDEL, invalid parameter delimiter - check use of special
characters \.Z\ [3]
$ SET PROCESS /PARSE_STYLE=EXTENDED [4]
$ OPEN /WRITE FILE z.z.z.z
$ [5]

1. The volume is ODS-5.
2. The parse style is set to TRADITIONAL.
3. DCL returns an error on some ODS-5 file names such as this one.
4. The parse style is set to EXTENDED.
5. DCL creates the file.
```

Examples of TRADITIONAL and EXTENDED styles on an ODS-2 volume:

```
Disk AABOUT$DKA200:, device type RZ25 Disk, is online, allocated, deallocate
on dismount, mounted, file-oriented device, shareable.

 Error count     0 Operations completed 232
  .
  .
  .
Volume Status: ODS-2, [1] subject to mount verification, file high-water
 marking, write-back caching XQP enabled, write-through XFC caching enabled.

$ SET PROCESS /PARSE_STYLE=TRADITIONAL [2]
$ OPEN /WRITE FILE z.z.z.z
%DCL-W-PARMDEL, invalid parameter delimiter - check use of special
characters \.Z\ [3]
$ SET PROCESS /PARSE_STYLE=EXTENDED [4]
$ OPEN /WRITE FILE z.z.z.z
%DCL-E-OPENIN, error opening
-RMS-E-CRE, ACP file create failed [5]
-SYSTEM-W-BADFILEVER, bad file version number

1. The volume is ODS-2.
2. The parse style is set to TRADITIONAL.
3. DCL returns an error message.
4. The parse style is set to EXTENDED.
5. DCL allows the file name, but XQP returns an error.
```

Examples of different error messages for the same syntax error:

```
$ SHOW DEVICE DKA500:/FULL
Disk AABOUT$DKA500:, device type RZ25 Disk, is online, allocated, deallocate
on dismount, mounted, file-oriented device, shareable.

 Error count     0 Operations completed 155
  .
  .
  .
Volume Status: ODS-5, [1] subject to mount verification, file high-water
marking, write-back XQP caching enabled, write-through XFC caching enabled.

$ SET PROCESS /PARSE_STYLE=TRADITIONAL [2]
$ CREATE a^<b.c
%DCL-W-PARMDEL, invalid parameter delimiter - check use of special
characters
\^\ [3]
$ SET PROCESS /PARSE_STYLE=EXTENDED [4]
$ CREATE a^<b.c
%CREATE-E-OPENOUT, error opening a^<b.c as output
-RMS-F-SYN, file specification syntax error [5]

1. The volume is ODS-5.
2. The parse style is set to TRADITIONAL.
3. DCL returns an error message for a syntax error.
4. The parse style is set to EXTENDED.
5. RMS returns a different error message for the same syntax error.
```

### 3.18.5.6 Performing Image Backups to Disk

In section 11.15.3, the following note should be appended to the end of the section:

BACKUP does not preserve GUID signature during image restore operation of the system disk on Integrity server systems. During restore, BACKUP calls SETBOOT to create a new GUID signature. Hence, during image restore operation BACKUP does not restore the original GUID signature, rather it creates a new GUID signature. As a result of this, Integrity servers system does not boot automatically from a disk created through an image restore operation.

If required to boot an Integrity servers system from a disk created through an image restore operation, you need to follow one of the method described below to update the GUID signature of the disk:

- Use the following procedure to add or validate the boot options:

  ```
  $ @SYS$MANAGER:BOOT_OPTIONS.COM
  ```

- Use the following command to update the boot block:

  ```
  $ SET BOOTBLOCK /INTEGRITY <destination_disk>:[VMS$COMMON.SYS$LDR]SYS$EFI.SYS
  ```

The following corrections pertain to the *HP OpenVMS System Manager's Manual, Volume 2: Tuning, Monitoring, and Complex Systems*.

### 3.18.5.7 Mounting a Volume With Caching Disabled

The following paragraphs should be appended to Section 4.4.

To disable XFC, enter the following command:

```
MOUNT/CACHE=NODATA
```

This command disables only data cache (XFC) while metadata cache (XQP) is enabled.

This example mounts a database volume labeled ORACLE_VOL1 with data cache (XFC) disabled:

```
$ MOUNT DUA100: ORACLE_VOL1 /CACHE = NODATA /SYSTEM
```

### 3.18.5.8 System-Wide Statistics

In section 4.5.6.1, the following changes should be made to the foot note:

[7] Reads bypassing cache — The total number of read I/Os since system startup that were seen by the cache but were not cached, for example, because they were too big, or they were for volumes mounted /NOCACHE or /CACHE=NODATA, or they specified one of the following QIO modifiers: IO$M_DATACHECK, IO$M_ INHRETRY, or IO$M_NOVCACHE.

[17] Write bypassing cache — The total number of write I/Os since system startup that were seen by the cache but were not cached, for example, because they were too big, or they were for volumes mounted /NOCACHE or/CACHE=NODATA, or they specified one of the following QIO modifiers: IO$M_DATACHECK, IO$M_ ERASE, IO$M_INHRETRY, or IO$M_NOVCACHE.

### 3.18.5.9 Disabling Caching for a Volume

In Chapter 4, a new section "Disabling Caching for a Volume" has to be added before Section 4.5.4, "Disabling Caching for a File".

The following text should be added to the "Disabling Caching for a Volume":

From OpenVMS Version 8.4 onwards, XFC can be dynamically enabled or disabled or cleared for a volume using the DCL "SET VOLUME" command. In the earlier versions, XFC caching attributes of the volume were specified when the volume was mounted. Once the volume is mounted there is no way to dynamically modify the XFC caching attributes. Therefore, to modify the XFC caching attributes, the volume had to be dismounted and mounted again with the appropriate XFC caching attributes.

With this feature, after the volume is mounted, you can modify the XFC caching attributes dynamically without dismounting and mounting the volume again.

| Use... | To... |
|---|---|
| SET VOLUME V1/CACHE=DATA | To Enable XFC caching for the volume V1 |
| SET VOLUME V1/CACHE=NODATA | To Disable XFC caching for the volume V1 |
| SET VOLUME V1/CACHE=CLEAR_ DATA | To Clear the contents of the volume V1 from cache |
| SET VOLUME V1/CACHE=(DATA,CLEAR_DATA) | To Enable XFC caching for the volume V1 and Clear the contents of volume V1 from the cache |
| SET VOLUME V1/CACHE=(NODATA,CLEAR_DATA) | To Disable XFC caching for the volume V1 and Clear the contents of volume V1 from the cache |
| SHOW MEM/CACHE=(VOL=V1) | To display the current XFC caching status of the volume V1 |

**Examples**

1.

```
$ SET VOLUME $DKA100/CACHE=CLEAR_DATA
```

This example clears the contents of the volume $DKA100 already present in the XFC cache. The caching attributes of the volume $DKA100 is not altered.

2.

```
$ SET VOLUME $DKA100/CACHE=DATA
```

This example enables XFC caching for the volume $DKA100. The contents of volume $DKA100 already present in the XFC cache is not affected.

3.

```
$ SET VOLUME $DKA100/CACHE=(DATA,CLEAR_DATA)
```

This example enables XFC caching for the volume $DKA100 and clears contents of the volume $DKA100 already present in the XFC cache.

### 3.18.5.10 Understanding File System Data Caches

In Section 4.2, add the following bullet after the following paragraph:

XFC improves I/O performance and contains the following features that are not available with VIOC:

- Dynamically enabling or disabling caching for mounted volumes

## 3.18.6 HP OpenVMS RTL Library (LIB$) Manual

*V8.3*

The LIB$SET_SYMBOL value-string is incorrectly documented in Version 8.2 of the *HP OpenVMS RTL Library (LIB$) Manual*. The correct value-string is as follows:

Trailing blanks are not removed from the value string before use. The maximum length of value-string is 4096 characters. Integer values are not allowed; LIB$SET_SYMBOL is intended to set string CLI symbols, not integer CLI symbols.

### 3.18.7 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.

### 3.18.8 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 out swapping 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."

### 3.18.9 *HP OpenVMS System Analysis Tools Manual*

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

### 3.18.10 *HP OpenVMS Programming Concepts Manual*

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

#### 3.18.10.1 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.

## 3.19 *HP OpenVMS DELTA/XDELTA Debugger Manual* Update

*V8.3*

The HP DELTA debugger was made available on OpenVMS Integrity servers 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 Integrity servers.

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

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

#### 3.19.1.1 /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 is as follows:

The /REMOVE qualifier has been enhanced for the Integrity servers 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.

#### 3.19.1.2 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.

## 3.19.2 *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*.

### 3.19.2.1 *HP OpenVMS RTL Library (LIB$) Manual*: Rounding Rule for LIB$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.

## 3.19.3 *HP OpenVMS RTL Library (LIB$) Manual*: Platform Restrictions

*V8.2–1*

The *HP OpenVMS RTL Library (LIB$) Manual* incorrectly identifies the following routines as being available on both Alpha and Integrity servers. 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 Integrity servers.

The routines relating to invocation contexts and invocation handles that are Integrity servers 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*.

### 3.19.4 *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 all Alpha and Integrity servers. 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 Integrity servers or on ES47 or GS1280 Alpha systems if you booted
from a Graphic console.
```

#### 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.

## 3.20 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 servers console and network boot drivers perform autonegotiation only. The network switch nearest to the Integrity servers boot client must be set to autonegotiate for a successful network boot. Failure to set the switch to autonegotiate may not complete the network boot process.

## 3.21 Synchronous Data Links not Supported

OpenVMS does not support any synchronous data link hardware on Integrity servers.

## 3.22 Duplex-Mode Mismatch Errors

*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.

# 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 8.4 New Features and Documentation Overview*.

## 4.1 SYS$TIMEZONE_RULE Logical Replaces Hyphen (-) with Caret (^)

*V8.4*

Starting from Version 8.2 onwards, SYS$TIMEZONE_RULE logical is modified to replace the '-' character with the '^' character. This change is done in TDF to support DTSS. DTSS cannot handle the commonly used UNIX 'GMT-X' timezone rules and does not support timezone rule strings that are identical to the timezone name.

For example, the 'GMT-1' timezone rule generates a SYS$TIMEZONE_RULE string of 'GMT-1'. Due to the matching rule file name of 'GMT-1' and rule string of 'GMT-1' caused DTSS not to function properly.

The CRTL and DTSS components are also modified to support this change.

For example, Timezone logical before this change:

```
"SYS$TIMEZONE_RULE" = "CET-1CEST-2,M3.5.0/02,M10.4.0/03"
```

Timezone logical after this change:

```
"SYS$TIMEZONE_RULE" = "CET^1CEST^2,M3.5.0/02,M10.4.0/03"
```

## 4.2 Issues with Time Zone Configuration

*V8.4*

On OpenVMS Version 8.4, if you configure time zone to a zone that does not get affected by the daylight saving time (DST) changes, it results in the following error message on the operator's terminal at a very high frequency:

```
%TDF-F-SMNSUBFAIL, Attempt to compute delta in Smithsonian time
failed, status = 001583EC, terminating.
```

**Note**

OpenVMS Version 8.4 upgrade kit is recommended only to those customers who will be configuring their time zone to a zone that gets affected by the DST changes.

#### Workaround

Deassign the SYS$DST_DELTA_TIME logical, when the error message appears on the system.

```
$ DEASSIGN/EXECUTIVE_MODE/SYSTEM SYS$DST_DELTA_TIME
```

## 4.3 OpenVMS as a Guest Operating System on Integrity VM

OpenVMS Version 8.4 now supports HP Virtualization and can be installed as a guest operating system on HP Integrity Virtual Machines (Integrity VM). For more information about product specific limitations, see the respective product documentation.

This section describes known problems and restrictions in OpenVMS guest on Integrity VM.

### 4.3.1 "Guest Punishment" Scenarios

*V8.4*

#### Scenario 1

A problem in Integrity VM Version 4.1 Field Test Evaluation Kit for OpenVMS Version 8.4 causes OpenVMS guests to fail with the following message displayed on the guest's MP console:

```
*** A fatal error has occurred -- VM terminated ***

**** Dumping Guest Image ****

**** Done with dump (nnnnnKbytes) ****
```

At the same time, the Integrity VM monitor log (found on the VM Host in /var/opt/hpvm/common/hpvm_mon_log) must include guest punishment information as follows:

```
Assertion failed mmxlate.c:nnn: PTE_TYPE(pte) == PTE_TYPE_TRPT || PTE_TYPE(pte) == PTE_TYPE_SPPT
scopeDeact EA0C000000000000
 ================================================================================

Guest punishment: Virtual Machine Monitor Assertion Failed  (CPUnnnnnnn)

================================================================================
```

This problem occurs when a privileged hardware instruction (for example, mov r1 = psr) is executed in process space, rather than in system space (that is, in the VMS executive).

When an application uses the $CMKRNL system service to enter kernel mode, it can execute privileged hardware instructions in process space. The kernel mode routines may be within the main image of the application, or within a shareable image. To run such an application, the user must have CMKRNL privilege, or the image must be installed: a main image with privileges, and a shareable image as a protected shareable image.

The application may be part of OpenVMS, a local customer application, or may be from a third-party.

If the problem occurs, a possible workaround is to install the image that performs the privileged hardware instructions using /RESIDENT. This ensures that the code of the image is in the system space.

### Examples

In the following examples, the image PUNISH_SHR.EXE is a protected shareable image that uses $CMKRNL to execute privileged hardware instructions. The image PUNISH_MAIN.EXE is an executable image that calls routines in PUNISH_SHR.EXE. The output is as seen at the guests MP console in each case.

a. The protected shareable image is installed without /RESIDENT:

```
$ set process/privilege=cmkrnl
$ install add sys$disk:[]punish_shr.exe/protect/share
$ set process/privilege=nocmkrnl
$ define/user punish_shr sys$disk:[]punish_shr.exe
$ run punish_main
Changing mode to kernel to read PSR

*** A fatal error has occurred -- VM terminated ***

**** Dumping Guest Image ****

**** Done with dump (22864Kbytes) ****
```

This is the VM Host view of the guest punishment in the Integrity VM monitor log at the same instant:

```
Assertion failed mmxlate.c:430: PTE_TYPE(pte) == PTE_TYPE_TRPT || PTE_TYPE(pte) == PTE_TYPE_SPPT
scopeDeact EA0C000000000000
================================================================================

Guest punishment: Virtual Machine Monitor Assertion Failed  (CPU2000000)

================================================================================
```

b. The protected shareable image is installed with /RESIDENT:

```
$ set process/privilege=cmkrnl
$ install add sys$disk:[]punish_shr.exe/protect/share/resident
$ set process/privilege=nocmkrnl
$ define/user punish_shr sys$disk:[]punish_shr.exe
$ run punish_main
Changing mode to kernel to read PSR
PSR = 000000100a0ae010
```

This problem will be fixed in a future version of Integrity VM.

### Scenario 2

OpenVMS guests will fail with the following message displayed on the guest's MP console:

```
*** A fatal error has occurred -- VM terminated ***

**** Dumping Guest Image ****

**** Done with dump (nnnnnKbytes) ****
```

The Integrity VM monitor log (found on the VM Host in /var/opt/hpvm/common/hpvm_mon_log) must include guest punishment information as follows:

```
ERROR: Could not allocate pinned memory (256 bytes)
Pinned pool: MaxChunk: 0x0000000000000000  Free: 0x0000000000000000
Pinned pool chunks:
Resource map: PinnedMemAlloc  limit=0xea0a0000025e0870
Assertion failed firmware.c:94: argArray

================================================================================

Guest punishment: Virtual Machine Monitor Assertion Failed  (CPUc000000)
```

==========

This problem occurs when VM Monitor runs out of memory to allocate for guests.

The workaround for this problem is to restart VM Monitor by entering the following commands on the Host:

```
# /sbin/init.d/hpvm stop
# /sbin/init.d/hpvm start
```

This problem will be fixed in a future version of Integrity VM.

### 4.3.2 Increased CPU Consumption After Shutdown

*V8.4*

OpenVMS guest has a known issue where the CPU consumption of the Host increases, in case the guest is shutdown using the SYS$SYSTEM:SHUTDOWN.COM command procedure using "NONE" as the shutdown options. The CPU consumption on the Host remains high as long as the VMS guest stays in the "P000>>>" firmware prompt. After the guest is rebooted the Host CPU consumption returns back to normal.

The suggested workaround for this problem is to shutdown VMS guest using "POWER_DOWN" as the shutdown options.

A known consequence of using this option is that, the virtual machine is shutdown and has to be restarted by the MP command "pc -on" in the virtual console or alternately enter the following command on the Host:

```
# hpvmstart -P <<OpenVMS guest name>>
```

This will be fixed in a future release.

### 4.3.3 OpenVMS Guest Does not Support Attached I/O Devices

*V8.4*

The OpenVMS guest does not support attached devices such as CD/DVD burners, media changers and tape devices. If you want to use tape devices, you can connect them to a physical system that is in a cluster with OpenVMS guest and TMSCP serves the tape devices.

### 4.3.4 Networking or Storage Interface Support

*V8.4*

The OpenVMS guest supports only Accelerated Virtual I/O (AVIO) interface.

Integrity VM commands allows you to configure VIO devices to a guest and these devices may not give any apparent errors during the startup. However, VIO devices are not part of the supported configuration of a guest running OpenVMS Operating System.

## 4.4 Provisioning OpenVMS Using HP SIM

The following release notes pertain to provisioning on OpenVMS.

### 4.4.1 Provisioning OpenVMS Guest Limitation

*V8.4*

Provisioning OpenVMS using HP SIM, Version 4.0 is not supported with OpenVMS as a Guest Operating System on Integrity VM.

### 4.4.2 System Firmware

*V8.4*

The system firmware version of BL860c and BL870c servers must be at 4.21. The system firmware version of rx3600 and rx6600 servers must be at 4.11.

### 4.4.3 Provisioning Multiple Servers

*V8.4*

- HP SIM provisioning using InfoServer can provision up to eight servers simultaneously.
- HP SIM provisioning using vMedia can provision only one server at a time.

### 4.4.4 Provisioning From HP SIM Central Management Server

*V8.4*

OpenVMS can be provisioned from an HP SIM Central Management Station, an HP ProLiant server running Microsoft Windows.

### 4.4.5 InfoServer Name Length

*V8.3-1H1*

The InfoServer name must be less than 12 characters long for provisioning to work. This is a temporary restriction.

### 4.4.6 OpenVMS InfoServer and the Integrity servers on the Same LAN

*V8.3-1H1*

The OpenVMS InfoServer and the Integrity servers must be on the same Local Area Network (LAN) to provision the server blade.

### 4.4.7 EFI Firmware

*V8.3-1H1*

The EFI firmware for the BladeSystem must be at version 5.0 or later.

### 4.4.8 Management Processor

*V8.4*

The Management Processor must be running the Advanced iLO2 firmware.

### 4.4.9 OpenVMS TCP/IP Provisioning Limitation

*V8.4*

The TCP/IP server components BIND, LPD, LBROKER, and SMTP, if selected to be enabled on the target server, do not start up when OpenVMS TCP/IP is configured through Provisioning.

The workaround for this problem is to configure and restart these services manually after configuring TCP/IP with Provisioning.

### 4.4.10 Limitation with Deploying OpenVMS on Multiple Target Servers Simultaneously

*V8.4*

There is a known issue with the TFTP server on Infoserver, which may prevent deploying OpenVMS Version 8.4 simultaneously on two or more target servers when using InfoServer booting with memory disk. In this scenario, InfoServer booting on the target server reports network errors when loading the memory disk similar to the below trace. These errors may prevent booting the target server successfully from InfoServer.

```
Shell> lanboot -dn sysmg3

Client MAC Address: 00 11 22 33 44 55
Client IP Address: 1.2.3.4
Subnet Mask: 255.255.254.0
BOOTP Server IP Address: 1.2.3.5
DHCP Server IP Address: 0.0.0.0
Boot file name: LDA30:[VMS$COMMON.SYSEXE]VMS_LOADER.EFI

Retrieving File Size.

Retrieving File (TFTP).Loading memory disk from IP 1.2.3.5
.
Warning - Unable to open SYS$LOADABLE_IMAGES:SYS$PLATFORM_SUPPORT.EXE, status = 0x54

*** SYSTEM MAY NOT BE BOOTABLE. Continuing...
..........

Warning - Unable to open SYS$SYSTEM:SYSBOOT.EXE, status = 0x54

*** SYSTEM MAY NOT BE BOOTABLE. Continuing...

...

Warning - Unable to open SYS$LOADABLE_IMAGES:EXCEPTION.EXE, status = 0x54

*** SYSTEM MAY NOT BE BOOTABLE. Continuing...
.

TFTP error, status = 8000000000000012, attempting retry
```

To workaround this issue, when deploying OpenVMS Version 8.4 simultaneously on two or more target servers with Provisioning, avoid using InfoServer boot with memory disk option as follows: In Step 2 of Provisioning in HPSIM GUI, under the "LAN Boot Settings" section of all target servers, select the setting for OpenVMS Version combo box as "V8.3-1H1 or below".

## 4.5 Insight Dynamics - Virtual Server Environment for OpenVMS

This section describes the known issues and limitations in Insight Dynamics - Virtual Server Environment (ID-VSE) Version 4.1 for OpenVMS.

### 4.5.1 Utilization Data Collection Fails

*V8.4*

When an attempt is made to collect utilization data in Capacity Advisor for an OpenVMS guest on Integrity VM, the operation fails and the following error message is displayed:

```
The system has no workload defined. Make sure to select
Tools->VSE Management...in HP-SIM before running this command
for the first time. For HPVM Guests, please be sure that the
HPVM wbem provider is properly configured.
```

#### Workaround

1. Click **Tools –> VSE Management...**. The **Virtualization Manager** page appears.
2. Select **OpenVMS guest**.
3. Select **Tools –> System Information –> System Page**. The **System Page...** appears.
4. Select the **Tools & Links** tab on the **System Page**, and then select **Edit System Properties**. The **Edit System Properties** page appears.
5. In the Product Description section, select the *Operating System for Tool Filtering* property. Click the drop-down menu and select *HP OpenVMS* and click **OK** to apply the changes.

   After making these changes, you can return to Capacity Advisor to collect and view utilization data profile for the OpenVMS guest.

### 4.5.2 Problem While Creating a New or Replacement Simulated System

*V8.4*

When trying to create a new or replacement simulated OpenVMS system in a Capacity Advisor Scenario, the *Select OS Type* in the "System Type and Size" section does not list "OpenVMS".

#### Workaround

For a new or replacement OpenVMS system, select the different operating system. To do so, follow these steps:

1. In the "System Type and Size" section, select the *Select OS Type* property.
2. Click the drop-down menu and select *HP-UX*.
3. Click **OK** to apply the changes.

### 4.5.3 Utilization Data not Available for OpenVMS Sub-OS Workloads

*V8.4*

The OpenVMS Utilization WBEM provider supports collecting utilization data only for OpenVMS whole-OS workloads. It does not support sub-OS (monitored and managed) workloads under OpenVMS. This has the following impact on ID-VSE:

- Virtualization Manager and Capacity Advisor display utilization data for whole-OS workloads only.
- Utilization data from sub-OS workloads running under OpenVMS cannot be used for capacity planning in Capacity Advisor.

### 4.5.4 Insight Software Features not Supported on OpenVMS

*V8.4*

The following Insight software features are not supported on OpenVMS:

- Application Discovery
- iCAP Manager
- Process Resource Manager (PRM)
- Logical Server Management

- Virtual Connect Enterprise Manager (VCEM)
- Partition Manager
- BladeSystem Integrated Manager
- GiCAP Group Manager
- Virtual Machines Manager
- Insight Power Manager (IPM)
- VSE troubleshooting for OpenVMS Managed Nodes - This refers to 'Check VSE CMS to Managed Node Communication...' and 'Check VSE Managed Node Configuration...' options under 'Diagnose' menu -> 'Troubleshoot VSE Management....'
- Configure VSE agents for OpenVMS - This refers to all options under the 'Configure' menu -> 'Configure VSE agents...
- VSE Agentless data collection - This refers to all options under the 'Configure' menu -> 'Configure VSE Agentless agents...
- Import of Capacity Advisor Data from OVPA and PMP - This refers to all options under Optimize menu -> Capacity Advisor -> Import Capacity Advisor Data

## 4.6 Performance Enhancements

*V8.4*

The following performance enhancements have been made to the OpenVMS Version 8.4 release.

### 4.6.1 Enhancements to Write Bitmaps

*V8.4*

Write Bitmaps (WBM) is a feature used by OpenVMS during minimerge and minicopy operations of Shadowing minimerge and minicopy. Information, about which blocks on a disk are written, is transmitted to other nodes within the cluster. The following updates have been made in this release.

#### 4.6.1.1 WBM_MSG_INT Parameter Updates

*V8.4*

The WBM_MSG_INT parameter indicates the time by which a SetBit message can be delayed when it is in buffered mode. If the SetBit buffer does not fill with SetBit messages by this time interval, then the message is sent. The parameter is in milliseconds, however, the conversion factor used for this timer was off by a factor of 10. Earlier, a WBM_MSG_INT value of 10 was resulting in a 100 millisecond delay when in buffered mode. This problem is corrected so that a value of 10 now indicates only a 10 millisecond delay.

#### 4.6.1.2 WBM_MSG_UPPER and WBM_MSG_LOWER Parameter Updates

*V8.4*

WBM_MSG_UPPER is the threshold used to determine if a switch should occur to buffered message mode, when operating in single message mode. If WBM_ MSG_UPPER or more SetBit operations are done in a 100 millisecond window, the messaging mode will be switched to buffered mode. The default value is 80.

WBM_MSG_LOWER is the threshold used to determine if a switch should occur to single message mode, when operating in buffered message mode. If WBM_ MSG_LOWER or fewer SetBit operations are done in a 100 millisecond window, the messaging mode will be switched to single mode. The default value is 20.

#### 4.6.1.3 Asynchronous SetBit Messages

*V8.4*

There can be multiple master bitmap nodes for a shadow set. Currently, SetBit messages are sent to the multiple master bitmap nodes synchronously. Only when the response for the SetBit message is received from the first remote master bitmap node, is the message sent to the next master bitmap node. When done with all of the remote master bitmap nodes, the I/O is resumed.

SetBit messages are now sent to all multiple master bitmap nodes asynchronously. I/O operation is resumed when the responses from all the master bitmap nodes are received. This reduces the stall time of the I/O operation by the write bitmap code.

#### 4.6.1.4 Reduced SetBit Messages for Sequential I/O

*V8.4*

If sequential writes occur to a disk, it results in sending Setbit messages that set sequential bits in the remote bitmap. The WBM code will now recognize where a number of prior bits in the bitmap have already been set. In this scenario, the WBM code will set additional bits so that if sequential writes should continue, fewer Setbit messages are required. Assuming the sequential I/O continues, the number of Setbit messages will be reduced by about a factor of 10 and thus improve the I/O rate for sequential writes.

### 4.6.2 Exception Handling Performance Improvements (Integrity servers Only)

*V8.4*

The OpenVMS Version 8.4 caches the decoded unwind data. The cache is used in the user-callable calling standard routines, during the exception handling. These calling standard routines are also used in the RTLs, to implement programming language constructs like the *try/throw/catch* constructs in C++ and the *setjmp/longjmp* constructs in C programming language.

In case of unexpected errors, the cache can be disabled temporarily using the VMS system parameter, *KTK_D3*. Its default value of zero enables the cache. A value of one disables the cache. The special parameter, *KTK_D3* may have been used by HP supplied debug/test images. If you had such test images on your system, make sure that it is reset to its default value zero. The ability to disable the cache will be removed in the OpenVMS Version 8.4 main release.

### 4.6.3 Exception Handling (Integrity servers Only)

Some performance improvements have been made to exception handling for OpenVMS Integrity server systems. The change will reduce the overhead of exception handling in some, but not all cases of exception handling.

### 4.6.4 Image Activation (Integrity servers Only)

During image activation and over the life of the image, paging IO brings pages of the image into memory. On Integrity server systems, an I-cache flush need to be performed on these pages in case the page has code that is executed. This resulted on the I-cache flush occurring on many pages that would never be executed. To avoid the I-cache flush on pages that are never executed, the I-cache is now only done on pages when an instruction is first executed on the page. This avoids the I-cache flush on the pages that are never executed and provides an overall system performance benefit.

### 4.6.5 Global Section Creation and Deletion

Performance improvements have been made to areas of the operating system that create and delete various types of global sections. The benefits of the changes will only be seen on large SMP systems as a reduction in MP Synch.

### 4.6.6 Dedicated CPU Lock Manager

The Dedicated CPU Lock Manager is a feature typically only turned used on systems with 16 or more CPUs and very high locking rates. Improvements have been made to the Dedicated CPU Lock Manager that results in an increase in the rate at which locking operations can be performed.

### 4.6.7 Ctrl/T Alignment Faults

A Ctrl/T operation at a terminal resulted in a number of alignment faults. These have been corrected for OpenVMS Version 8.4.

## 4.7 Error and Warning Messages from ACPI During Boot

*V8.4*

The following message may be displayed by VMS during boot on cell-based machines (for example, rx8640 or rx7640):

```
ACPI Error (utmutex-0430): Mutex [1] is not acquired, cannot release [20071219]
```

The following message may be displayed by VMS during boot on certain systems that have power management enabled (for example, an rx2660 with the latest processors):

```
ACPI Warning (nseval-0250): Excess arguments - method [_OST] needs 3, found 7 [20080701]
```

These messages can be ignored. They will be fixed in a future release.

## 4.8 Large Device Name Support for Accounting Utility

*V8.4*

The accounting utility is modified to handle long device names. It can now display device names having seven characters or more, for example, Terminal (TNA) of unit number >9999, MBA device of unit number >999, and other large device names such as TNA10000:, MBA1000:, and so on.

Earlier, the utility displayed arbitrary characters if a device name exceeded seven characters. A new accounting record version (version4) is used to write new records into the accounting.dat file and the utility is modified appropriately to read and display these new records.

# 4.9 PAGED_LAL_SIZE New System Parameter

PAGED_LAL_SIZE sets the maximum size, in bytes, to use the page dynamic pool lookaside lists.

## 4.9.1 Paged Pool Lookaside Lists

*V8.4*

Paged dynamic pool now allows the use of lookaside lists to increase system performance in some cases. It is controlled by SYSGEN parameter PAGED_LAL_SIZE and is off (0) by default.

If the variable paged pool freelist becomes fragmented, you might benefit by enabling the use of these lookaside lists. The SYSGEN parameter PAGED_LAL_SIZE sets the maximum size, in bytes, to use for these lookaside lists. Packets larger than this size will still be allocated from the variable paged pool freelist. A modest value, 512 bytes, has been found to help systems doing intensive logical name creation and deletion operations.

Since the parameter is dynamic it can be enabled, adjusted, or disabled as needed. If it was enabled and then lowered there might be some packets on the paged pool lookaside lists that are no longer actively in use. These show up as "Over-limit Lookaside Blocks" in DCL's and SDA's `SHOW MEMORY/POOL/FULL` command. These packets were used before but are now larger than the new PAGED_LAL_SIZE. These packets will be used again if the SYSGEN parameter is increased to include them, or if there is a paged pool shortage and the packets are reclaimed from the lookaside lists.

To help prevent a runaway condition where packets on a lookaside list starts to consume most or all of paged pool, the paged pool lookaside lists will not be used for packets in the last quarter of paged dynamic pool. If there is a paged pool memory shortage packets on the lookaside lists will be reclaimed as well.

If disabled, at the default value of 0, paged pool behaves as it did in previous versions of OpenVMS, allocating and deallocating packets from the paged pool variable freelist.

# 4.10 2 TiB Disk Volume Support Restrictions

*V8.4*

OpenVMS Version 8.4 supports disk volumes up to 2 TiB in size with the following restrictions:

- With OpenVMS versions prior to version 8.4, there is no support for volumes larger than 1 TiB in size or for mounting of volumes larger than 1 TiB. To prevent accidental mounts on earlier versions of OpenVMS, the latest patches for MOUNT will explicitly disallow mounting of volumes larger than 1 TiB on such systems.
- The lexical function F$GETDVI( ) with items codes MAXBLOCK, FREEBLOCKS, EXPSIZE, and VOLSIZE will return a negative number if the value is larger than 1 TiB. This is due to the fact that DCL does 32-bit signed integer arithmetic and comparisons. Command procedures that use F$GETDVI( ) with these item codes may need to be modified to work with volumes larger than 1 TiB. For more information about handling numeric values outside the range of DCL integer representation using DCL, see the *HP OpenVMS DCL Dictionary*.

## 4.11 Configuring SAS Tape Drives

*V8.4*

SAS tape drives must be named and configured using the same commands that are used to configure Fibre Channel tape drives. For more information, see the section 7.5 "Fibre Channel Tape Support" in the *Guidelines for OpenVMS Cluster Configurations*.

## 4.12 External SAS Disk Device Naming

*V8.4*

The external SAS drives that are served by non-Smart array controllers can be configured as $3$DGA<UDID>, where UDID is unique device ID for the LUN. Note that Fibre Channel disk device names use an allocation class value of 1 whereas external SAS disk device names use an allocation class value of 3 to differentiate a SAS device from an Fibre Channel device.

## 4.13 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*.

**Note**

A special note for external authentication users.

If you are using the SYS$ACM-enabled LOGINOUT.EXE and SETP0.EXE (SET PASSWORD) images that supports external authentication, an upgrade to higher version of OpenVMS will restore the setup.

If you are using the password policy for customized password processing, it is necessary to restart the ACME Server after the Password Policy shareable image is installed, and the LOAD_PWD_POLICY system parameter is enabled.

Please see the SYS$HELP:ACME_DEV_README.TXT on how to install the ACMELOGIN kit.

### 4.13.1 External Authentication and Password Policy

*V8.4*

If you are using external authentication to authenticate users against a source other than the SYSUAF.DAT, and using the password policy for customized password processing, it is necessary to restart the ACME Server after the Password Policy shareable image is installed, and the LOAD_PWD_POLICY system parameter is enabled.

Use the following command to restart the ACME Server:

```
$ SET SERVER ACME_SERVER /RESTART
```

### 4.13.2 Integrity servers External Authentication Support

*V8.2*

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

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

### 4.13.3 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
$
```

### 4.13.4 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.

### 4.13.5 Restriction in ACME_SERVER Process (Integrity servers only)

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

**Note**

Do not increase the number of threads on Integrity servers. Increasing the number of threads on Integrity servers might lead to ACME_SERVER process crash and login failures.

## 4.14 Itanium Primary Bootstrap (IPB) Fails to Find the Valid Dump Devices

*V8.4*

Connecting a bridged device such as, AD221, HP PCIe combo Card on the PCI bus, where dump devices (DOSD) are configured on another HBA that is already connected may cause the PCI bus numbering of the dump devices to be renumbered and making it difficult to find the valid dump devices.

**Workaround**

After connecting a new I/O card, validate the boot/dump option. Then, refresh the DUMP_DEV and boot device list.

## 4.15 SHUTDOWN.COM Changes

*V8.4*

`SHUTDOWN.COM` is modified to execute a pre-queue system shutdown procedure SYSHUTDWN_0010.COM if it is present. The template contains three sample routines that can help force the queue system to shutdown and restart or failover faster.

## 4.16 OpenVMS Cluster Systems

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

### 4.16.1 Cluster over IP (IP Cluster Interconnect)

HP OpenVMS Version 8.4 is enhanced with the Cluster over IP feature. This feature provides the ability to form clusters beyond a single LAN or VLAN segment using industry standard Internet protocol. It also provides improved disaster tolerant capability to OpenVMS clusters.

This section describes known problems and restrictions in Cluster over IP.

#### 4.16.1.1 Software Requirements

*V8.4*

Cluster over IP is available only on OpenVMS Version 8.4 Alpha and Integrity servers. Cluster over IP also requires HP TCP/IP services for OpenVMS, Version 5.7.

#### 4.16.1.2 Integrity servers Satellite Node and Bootserver in the Same LAN

*V8.4*

An Integrity server satellite node must be in the same LAN as its boot server for the satellite node to initialize cluster over IP successfully and to join the cluster successfully.

It is also necessary to have LAN cluster communication between Integrity servers satellite node and the boot server for the satellite node to be able to initialize cluster over IP during the satellite bootup.

#### 4.16.1.3 Alpha Satellite Node Requires LAN Channels With Disk Server

*V8.4*

Alpha satellite boot fails in an IP only environment. That is, while booting an Alpha satellite, if all the nodes, including the boot servers, are using only IP channels for cluster communication, the satellite boot fails with the following message:

```
cluster-W-PROTOCOL_TIMEOUT, NISCA protocol timeout %VMScluster-I-REINIT_WAIT,
Waiting for access to the system disk server
```

#### 4.16.1.4 IPv6 Support

*V8.4*

Cluster over IP does not support IPv6 type address for cluster communication interface.

#### 4.16.1.5 Dynamic Host Configuration Protocol (DHCP) or Secondary Address Support

*V8.4*

Cluster over IP requires the addresses that are used for cluster communication, which are static, primary address on that interface. Furthermore, IP address used for cluster communication must not be used for Failsafe configuration.

#### 4.16.1.6 Multiple IP Interface Configuration

*V8.4*

If you configure multiple IP interface with the same default gateway, loss of communication on any interface may result in disrupted cluster communication with CLUEXITS.

#### 4.16.1.7 ifconfig Command Usage

*V8.4*

If the interface used for cluster communication is reactivated by ifconfig, it results in losing cluster communication to other nodes and also results in cluexit of nodes.

#### 4.16.1.8 Multiple Gateway Configuration

*V8.4*

Cluster over IP configuration information is stored in the configuration files, which are loaded early in the boot time. This configuration information also includes the default route or gateway that is used by TCP/IP. Currently, only one default route can be entered in the configuration file and used during the node bootup.

#### 4.16.1.9 Block Transfer XMIT Chaining

*V8.4*

PEdriver emulates each IP interface used for cluster communication similar to lan interface (BUS). An IP bus will have the characteristics of Xchain_Disabled status as shown. This means that the block transfer packets transmitted through TCP/IP are copied from the PEdriver to the TCP/IP buffers.

```
$ mc scacp show ip

NODEG PEA0 Device Summary 16-FEB-2009 12:29:15.92:

        Device  Errors +                 Mgt   Buffer  MgtMax   Line     Total     Current
 Device  Type    Events   Status       Priority  Size   BufSiz  Speed   Pkts(S+R)  IP Address
 ------  ----    ------   ------       --------  -----   ------  -----  ---------  -----------
  IE0             184  Run Online             0   1394        0    N/A    1419711  15.146.235.222
                       XChain_Disabled
```

#### 4.16.1.10 LANCP for Downline Load

*V8.4*

Cluster over IP requires LANCP, instead of DECnet for downline load since the changes related to configuring cluster over IP and enabling cluster over IP is available only with CLUSTER_CONFIG_LAN.COM. This restriction will be fixed in the future release of HP Clusters.

#### 4.16.1.11 Duplex Mismatch

*V8.4*

A duplex mode mismatch or a change in duplex mode from half to full on the host duplex can result in CLUEXIT when IP is used for cluster communication. It is recommended to check for the duplex mismatch issues to avoid cluexit.

#### 4.16.1.12 Configuring a Node During Upgrade

*V8.4*

If you are upgrading from prior versions to Version 8.4, you cannot enable Cluster over IP. When upgrading it does not call CLUSTER_CONFIG[_LAN] procedure, which is required for enabling Cluster over IP. Hence, the node joins the existing cluster in which it is the member before upgrading.

For enabling Cluster over IP, you must call CLUSTER_CONFIG_LAN procedure explicitly after upgrading.

This restriction will be removed in a future release.

### 4.16.2 OpenVMS Cluster Support for Integrity VM

*V8.4*

OpenVMS for Integrity servers Version 8.4 is supported as a guest operating system on Integrity VM. OpenVMS guest can be configured in a cluster.

#### 4.16.2.1 Cluster Interconnect for OpenVMS Guest

*V8.4*

OpenVMS guest can use both LAN or Cluster over IP (IPCI) to communicate with other nodes in the cluster.

#### 4.16.2.2 MSCP Support for Clusters in Integrity VM Environment

*V8.4*

MSCP is used to provide shared storage capability in cluster consisting of OpenVMS guest systems.

#### 4.16.2.3 Online Migration Support

*V8.4*

Online migration of OpenVMS guest that are part of cluster is not supported.

### 4.16.3 Mixed Platform Support

*V8.2*

- A supported production cluster containing an Integrity servers 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 Integrity servers 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.

### 4.16.4 Satellite Systems using Port Allocation Class

*V8.2*

Integrity server 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 *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.

## 4.17 Mixed-version Cluster Compatibility of a Six-member Shadowset

*V8.4*

OpenVMS Version 8.4 supports "Extended Membership" volume shadowing feature. This feature allows shadowsets to have more than three and up to six-members. This feature is enabled when a fourth member is added to the shadowset. Following are some of the important points in a mixed-version OpenVMS cluster:

- To use the "Extended Membership" shadowing feature, all the systems that mount the shadowset must be running OpenVMS Version 8.4.
- If you attempt to mount a shadowset on an OpenVMS Version 8.4 system using "Extended Memberships" shadowing feature, the mount fails if the shadowset is already mounted on systems with prior versions of OpenVMS in the cluster.
- If you attempt to mount a shadowset on a system that is not capable of "Extended Memberships" shadowing feature on prior versions of OpenVMS, the mount fails if shadowset is already mounted on an OpenVMS Version 8.4 system in the cluster using the "Extended Memberships" shadowing feature.

- Once the shadowset has been enabled to use "Extended Memberships" shadowing feature, the characteristic is maintained even if the membership is reduced to less than four members. The characteristic is retained until the shadowset is dismounted clusterwide.
- This shadowing feature is not ported onto OpenVMS VAX. If a shadowset is mounted on OpenVMS Alpha or OpenVMS Integrity servers without enabling this feature, the shadowset will mount on the OpenVMS VAX systems. The Virtual Unit characteristic voting ensures compatibility.

## 4.18 Backward Compatibility of a Six-member Shadowset

*V8.4*

A new area of the Storage Control Block (SCB) of disk stores the extended membership arrays required to support the "Extended Membership" shadowing feature. Therefore, an attempt to mount a six-member shadowset on prior versions of OpenVMS works only if the members are specified in the command line.

In OpenVMS prior versions, the $MOUNT/INCLUDE qualifier which is used for reconstructing the shadowset, can find only the existing membership list and not the new membership area in the SCB. Hence, it does not mount any members from the new extended membership area in the SCB.

This problem has been fixed in OpenVMS Version 8.4 upgrade kit.

## 4.19 WBEM Services and WBEM Providers for OpenVMS

This section describes known problems and restrictions in WBEM.

### 4.19.1 Increased CPU Consumption With WBEM on OpenVMS Guest

*V8.4*

OpenVMS Guest has a known issue where the CPU consumption of the host gradually increases with the time when WBEM is configured and running on the guest. Due to this issue, the guest responsiveness gradually decreases with the time, although there is no workload on the guest.

The workaround for this problem is to stop and restart WBEM on the guest when responsiveness is slow by entering the following command:

```
$ @SYS$STARTUP:WBEM_SERVICES$SHUTDOWN     ! Shutdown WBEM on the guest
$ @SYS$STARTUP:WBEM_SERVICES$STARTUP      ! Startup WBEM on the guest
```

Alternately, you can reboot the OpenVMS guest when the responsiveness is slow.

### 4.19.2 WBEM Providers Support for OpenVMS Guest

*V8.4*

WBEM Providers running on OpenVMS guest does not support WBEM instance data and event indications for CPU, Memory, Enclosure, Chassis, Fan, Power Supply, and Management Processor, since the guest is running in a virtual machine. This will be supported by WBEM providers running on the underlying VM Host operating system.

### 4.19.3 Based on OpenPegasus 2.9

WBEM Services for OpenVMS Version 2.9 is based on the OpenPegasus 2.9 code stream of The Open Group's Pegasus open source project.

### 4.19.4 Supports nPartitions and iCAP

On cell-based systems, Version 2.0 supports local nPartitions and iCAP providers. Only the functions and capabilities needed by these providers are supported.

### 4.19.5 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.)

### 4.19.6 Use Quotes Around Command Line Options

Ensure that you use quotes around a command line option to preserve its case. For example,
Correct:

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

Incorrect:

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

## 4.20 Writing the System Dump File to an Alternate Disk

*V8.4*

On Superdome class of servers, writing the system dump file to an alternate disk (DOSD) does not work and the following message is displayed:

```
**** Unable to locate SYSDUMP.DMP on any valid DUMP_DEV device
**** Attempting to write the crash dump to the system disk
```

## 4.21 Monitor Utility Changes

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.

### 4.21.1 Guest Operating System on Integrity VM

*V8.4*

OpenVMS Integrity servers Version 8.4 supports Guest Operating System on HP Integrity Virtual Machines (Integrity VM). When the OpenVMS Integrity servers is running as a guest on an Integrity VM system, the monitor utility indicates the amount of CPU time used by the guest. The Monitor also indicates the amount of CPU time allocated to the guest by Integrity VM.

The `MONITOR MODES` and `MONITOR SYSTEM /ALL` commands provide this information. When the system is running as a guest, the above commands display "In use by Host" instead of "Compatibility Mode". This field is to be interpreted as the amount of CPU time that was unavailable to the current guest and that is being used by the other guests or Integrity VM. The display is scaled based on the number of vCPUs (Virtual CPUs) configured for the guest irrespective of the actual number of physical CPUs in the host.

## System Management Release Notes
## 4.21 Monitor Utility Changes

```
$ MONITOR MODES
                        OpenVMS Monitor Utility
            +-----+     TIME IN PROCESSOR MODES
            | CUR |          on node VMSG7
            +-----+      5-FEB-2009 12:35:39.74

                                 0         25        50        75       100
                                 + - - - - + - - - - + - - - - + - - - - +
 Interrupt State                 |
                                 |         |         |         |         |
 MP Synchronization              |
                                 |         |         |         |         |
 Kernel Mode                     |
                                 |         |         |         |         |
 Executive Mode                  |
                                 |         |         |         |         |
 Supervisor Mode                 |
                                 |         |         |         |         |
 User Mode                    99 |||||||||||||||||||||||||||||||||||||||||
                                 |         |         |         |         |
 In use By Host                1 |
                                 |         |         |         |         |
 Idle Time                       |
                                 + - - - - + - - - - + - - - - + - - - - +
```

```
$ MONITOR SYSTEM/ALL
                        OpenVMS Monitor Utility
                           SYSTEM STATISTICS
                             on node VMSG9
                        5-FEB-2009 12:36:44.88

                                    CUR        AVE        MIN        MAX
    Interrupt State                0.00       0.12       0.00       0.33
    MP Synchronization             0.00       0.00       0.00       0.00
    Kernel Mode                    0.00       0.06       0.00       0.50
    Executive Mode                 0.00       0.00       0.00       0.00
    Supervisor Mode                0.00       0.00       0.00       0.00
    User Mode                     98.33      98.03      96.50      98.50
    In use By Host                 1.66       1.77       1.33       3.33
    Idle Time                      0.00       0.00       0.00       0.00
    Process Count                 25.00      24.72      24.00      25.00
    Page Fault Rate                0.00      10.96       0.00      47.50
    Page Read I/O Rate             0.00       0.96       0.00       3.16
    Free List Size             46851.00   46945.54   46850.00   47105.00
    Modified List Size           317.00     316.90     316.00     317.00
    Direct I/O Rate                0.00       1.37       0.00       5.50
    Buffered I/O Rate              1.00       2.68       0.66       9.83
```

**Note**

The data that is displayed when `MONITOR MODES` and `MONITOR SYSTEM /ALL` commands are executed on a guest is the time that the guest spends on the virtual CPUs.

### 4.21.2 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.

### 4.21.3 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 some 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] | MON31050 |
| OpenVMS Versions 8.2, 8.2-1 with remedial kit[1] | MON01060 |
| OpenVMS Versions 8.3, 8.3-1H1, 8.4 | 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.

## 4.22 System Parameters

*V8.3-1H1*

This release also contains the new GH_RES_CODE_S2 parameter, which specifies the size in pages of the resident 64-bit S2 space resident image code granularity hint region.

Only images linked with the /SEGMENT=CODE=P2 qualifier can have code placed in this region. See the *HP OpenVMS Linker Utility Manual* and the INSTALL utility in the *HP OpenVMS System Manager's Manual* for more information.

GH_RES_CODE has the AUTOGEN and FEEDBACK attributes.

## 4.23 SYS$LDDRIVER Restriction

*V8.3-1H1*

SYS$LDDRIVER.EXE is a freeware pseudo device driver that allows OpenVMS operating system to create virtual disks. For OpenVMS Version 7.3-1 and succeeding versions, this driver was placed in SYS$COMMON:[SYS$LDR] to support the creation of the source virtual disk for mastering a CD or DVD using CDRECORD or COPY/RECORDABLE_MEDIA. This is the only supported use of this freeware driver. All other uses of this driver continue to be subject to the following documented freeware usage restrictions:

OpenVMS Freeware is provided as is without a warranty. HP imposes no restrictions on its distribution or redistribution. HP does not support services for this software, fix the software, or guarantee that it works correctly.

## 4.24 CPU_POWER_MGMT Default Value Changed

*V8.3-1H1*

The default value for the sysgen parameter CPU_POWER_MGMT has been restored to 1 (that is to on). An improved idle power saving algorithm reduces interrupt latency while CPU_POWER_MGMT is on.

## 4.25 Booting A Satellite System with Reserved Memory

*V8.3-1H1*

To use the SYSMAN reserved memory feature on an Integrity server satellite system the file SYS$SYSTEM:VMS$RESERVED_MEMORY.DATA must allow world READ+EXECUTE access. Failure to set this access protection results in the warning when booting the satellite:

```
%VMS_LOADER-W-Warning: Unable to load file SYS$SYSTEM:VMS$RESERVED_MEMORY.DATA
```

After running SYSMAN to add memory reservations to a satellite, execute SYS$MANAGER:CLUSTER_CONFIG_LAN.COM to set the correct protection on the VMS$RESERVED_MEMORY.DATA file. To set the protection, from the cluster configuration procedure "Main Menu" select:

```
3. CHANGE a cluster member's characteristics.
```

From the "CHANGE Menu" select the following:

```
13. Reset an IA64 satellite node's boot environment file protections.

    What is the satellite name (leave blank to use a specific device and root)?
```

Enter the satellite name or satellite boot device and root for the system where you added the memory reservation. SYSMAN will be fixed in a later release to eliminate this condition.

## 4.26 SCACP Error Counter Reports Retransmit Errors

*V8.3-1H1*

If the PEA0: device on the system shows a number of errors, these errors might be retransmit errors and not actual errors. To verify actual errors, use the SCACP utility to confirm whether there are a number of retransmits on the PEA0 channels and use the LANCP utility to identify whether any actual devices errors

exist on the LAN devices that the PEdriver uses. If there are retransmits and no devices errors, then the PEA0: device errors are likely retransmits and not actual errors.

## 4.27 Virtual Connect

The following section pertain to Virtual Connect.

### 4.27.1 Failover and RECNXINTERVAL

*V8.3-1H1*

RECNXINTERVAL may have to be increased above the default of 20 to allow time for Virtual Connect Manager failovers. This is especially true in larger clusters.

## 4.28 INITIALIZE/ERASE=INIT Before Using Media

*V8.3-1H1*

HP recommends that you issue the DCL command INITIALIZE/ERASE=INIT on storage media prior to using them for the first time. This eliminates any stale data that was left from previous use by another operating system or diagnostics.

An indication of such stale data is three questions marks (???) in the console command output, as shown in the following example:

```
Shell> ls fs1:\
Directory of: fs1:\
 00/00/07 19:16p     1,788,984,016 ???
 00/00/80 12:00a               0 ???
     2 File(s)  1,788,984,016 bytes
     0 Dir(s)
```

The problem will be corrected in a future release.

## 4.29 Performance Data Collector for OpenVMS (TDC)

*V8.3-1H1*

TDC_RT Version 2.2-107 is included in the OpenVMS Version 8.3–1H1 installation. An update to TDC Version 2.2-108 is now available from the TDC Web site at:

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

TDC Version 2.2-108 corrects several issues discovered in TDC_RT Version 2.2-107. It also enables collection of internet metrics in TCPware and MultiNet environments, adds additional metrics to several data records, and provides new programming features and sample code in the TDC Software Developers Kit.

## 4.30 Recovering From System Hangs or Crashes (Integrity servers 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 a second time 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.

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

*V7.3-2*

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

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

## 4.32 Device Unit Number 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.

## 4.33 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.

## 4.34 Using EFI$CP Utility 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 Integrity servers unbootable.

## 4.35 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.

## 4.36 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 Integrity servers Version 8.2** | |
| | VMS82I_UPDATE-V0200 |

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

## 4.36.1 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://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 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 4.16.3.

Note that VAX systems cannot be in a cluster with Integrity servers. For a complete list of warranted groupings within a cluster, refer to the *HP OpenVMS Version 8.3 Upgrade and Installation Manual*.

### 4.36.2 API to Correct Incompatibility of FC 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*.

### 4.36.3 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.

### 4.36.4 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 or 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.

### 4.36.5 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.

### 4.36.6 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.

## 4.37 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.

### 4.37.1 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. |

## 4.38 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.

### 4.38.1 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.

### 4.38.2 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.

### 4.38.3 Galaxy on ES40: Turning Off Fastpath

*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 Fastpath 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.

## 4.39 Corrupted 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.

## 4.40 System Parameters

The following sections contain notes related to system parameters.

### 4.40.1 New System Parameters

*V8.3*

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

### 4.40.2 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
- TTY_SILOTIME
- BALSETCNT
- BREAKPOINTS

- MMG_CTLFLAGS
- MULTITHREAD
- NISCS_MAX_PKTSZ
- NISCS_PORT_SERV
- SECURITY POLICY

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.

### 4.40.3 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 - Integrity servers support added
- NISCS_MAX_PKTSZ - wording changes
- NISCS_PORT_SERV - bit definition changes
- SECURITY POLICY - Bits 13 and 14 defined
- 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*.

## 4.41 Terminal Fallback Facility

*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.

## 4.42 User Environment Test Package (Integrity servers 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.

## 4.43 Recommended Caching Methods

*Permanent Restriction*

Virtual I/O Cache (VIOC) — also known as VAX Cluster Cache (VCC) — is not available on OpenVMS Integrity servers. On Integrity servers, 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 Integrity servers. 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.

## 4.44 Analyze Utility for OpenVMS

The following sections describe corrected problems in the Analyze Utility for OpenVMS.

### 4.44.1 Formatted Symbol Vector Correctly Shown in Data Segment

Previously, the symbol vector summary information did not indicate the segment in which the symbol vector resided. The symbol vector was formatted only in the dynamic segment.

This problem is fixed in OpenVMS Version 8.3-1H1. The formatted symbol vector now appears with the data segment in which it is contained. The formatted symbol vector is embedded in data and visible in a dump of the data.

To avoid formatting the same data twice, the symbol vector is no longer shown with the dynamic segment. To make formatting of the symbol vector easy, the SYMBOL_VECTOR keyword is allowed for the /SEGMENT qualifier. When you specify this keyword, the resulting output is only the formatted symbol vector. The surrounding data are not shown. To show and format all of the data, select the segment by number.

To get equivalent output for the former command /SEGMENT=DYNAMIC for symbol vectors, use the /SEGMENT=(DYNAMIC,SYMBOL_VECTOR) qualifier.

The summary information shows the name of the data segment that contains the symbol vector.

### 4.44.2 Transfer Array Formatted in Data Segment

Previously, if you selected the data segment that contained the transfer array (either by segment number or with the ALL keyword), the transfer array was not formatted. Information about the transfer array was shown only in the summary.

This problem is corrected in OpenVMS Version 8.3-1H1. The formatted tranfer array now appears in the data segment.

### 4.44.3 System Version Array Formatted in Dynamic Segment

System version data is in the dynamic segment. Previously, if you selected the dynamic segment (either by segment number, or with the ALL or DYNAMIC keyword), the system version array was not shown. Information about the system version array was only shown in the summary.

This problem is corrected in OpenVMS Version 8.3-1H1. The formatted system version array now appears in the dynamic segment.

### 4.44.4 Enhancements for the /SEGMENT Qualifier

Enhancements have been made to the /SEGMENT qualifier for dynamic segments. Analyze has been enhanced to accept keywords for the /SEGMENT=DYNAMIC qualifier to provide customized information. The keywords for selectable information are:

- ALL—(Default) Formats all parts of the dynamic segment
- TAGS—Formats the tag array
- IMAGE_STRINGS—Formats strings of the specified image
- RELOCATIONS—Formats the image relocations
- FIXUPS—Formats the image fixups
- SYSTEM_VERSION_ARRAY—Formats the system version array

The default, /SEGMENT=ALL, formats all of the image information.

Note that formatting using the TAGS keyword includes the names of the needed images, so you do not have to add IMAGE_STRINGS to print the names.

### 4.44.5 Support for Section Escaping Added

On OpenVMS V8.3, the Analyze utility did not complete when analyzing an object module with more than 65,280 sections. Instead, it looped when attempting to print the section header table.

This problem has been fixed in OpenVMS V8.3-1H1.

## 4.45 INSTALL Utility for OpenVMS (Installing Resident Images in S2 Space)

The INSTALL utility now supports installing code segments of resident images into 64-bit S2 address space. Not all code can run in a full 64-bit address space (P2 or S2). For example, the code must be prepared for 64-bit PCs when handling exceptions. Also, some compilers require the /POINTER_SIZE=64 command qualifier, when generating code, suitable for a 64-bit address space.

To avoid mapping unprepared code in S2 space, the INSTALL utility by default will continue to map the code segments in S0/S1 space. The INSTALL utility will map code segments of resident images to S2 if two conditions are met:

- The developer explicitly confirmed that the code is 64-bit ready by using the /SEGMENT_ATTRIBUTE=CODE=P2 qualifier when linking the image.
- There is sufficient pre-allocated space in the resident code region in the S2 space to map the code segments. The size of the region is determined by the system parameter GH_RES_CODE_S2 (number of pages). The default value is set to 0. That means that by default even 64-bit ready resident images have their code mapped in S0/S1 space.

# 5
# Programming Release Notes

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

## 5.1 Symbolic Debugger

*V8.4*

On OpenVMS Alpha Version 8.4, when you set breakpoints the debugger will not be able to differentiate between FORTRAN functions and declared variables of the same name in different compilation unit.

## 5.2 C++ Run-Time Library

*V8.3-1H1*

Problems corrected in OpenVMS Version 8.3-1H1 include the following:

- The run-time library had faulty code that accessed memory just freed to advance a pointer. In multithreaded code, another thread could reuse that memory before the original thread could advance its pointer. This has been fixed by updating accesses prior to freeing pointers.
- A new processwide exception processing mode—`pure_unix`— has been introduced. In this mode, non-C++ exceptions, also known as OpenVMS conditions, cannot be caught in a C++ catch-all handler. This mode can be requested by calling `cxxl$set_condition(condition_behavior)` with a `pure_unix` argument:

  ```
  cxxl$set_condition(pure_unix);
  ```

  ```
  condition_behavior enum declared in <cxx_exception.h> header
  has been extended to include pure_unix member.
  ```

  To demonstrate how `pure_unix` mode works, consider the following program sample. As it is written, the program crashes with ACCVIO. If the call to `cxxl$set_condition()` is commented out, the program outputs "caught" and exits:

  ```
  #include <stdio.h>
  #include <cxx_exception.h>
  ```

  ```
  void generateACCVIO() { *((int*)0) = 0; }
  ```

```
int main() {
cxxl$set_condition(pure_unix);
try {
generateACCVIO();
             }
catch(...) {
puts("caught");
                    }
                 }
```

**Note**

To use this new functionality you must have a new version of `cxx_exception.h`, which is included in the CXXL$ANSI_DEF.TLB file provided with the Version 7.3 compiler (or higher).

- The run-time library sometimes failed to destruct objects of automatic storage duration defined in a function, if such a function exited via an exception that could be caught. This problem has been fixed.
- The run-time library now allows the thread cancel signal (CMA$_ALERTED) and the thread exit signal (CMA$_EXIT_THREAD) to be caught in a catch handler with a pointer or a reference to type CXXL$PTHREAD_CANCEL (or CX6L$PTHREAD_CANCEL) and CXXL$PTHREAD_EXIT (or CX6L$PTHREAD_EXIT), respectively, if catching the signals are enabled. The new types catch these signals exclusively.

**Note**

To use this new functionality, you must have a new version of `cxx_exception.h`, which is included in the CXXL$ANSI_DEF.TLB provided with the V7.3 compiler (or higher).

- The C++ RTL has changed its internal mapping of SIGTRAP from SS$_BREAK to SS$_TBIT, to match a recent C RTL change.
- The C++ RTL used to call `std::terminate()` when a destructor raised an exception during stack unwinding, even if that destructor did not exit via the exception. This problem has been fixed.
- The C++ RTL used to call `std::terminate()`, if a foreign exception (such as a non-C++ OpenVMS condition) was raised while a C++ exception was being processed. This behavior has been refined to calling `std::terminate()` only if the raised OpenVMS condition also leads to unwinding the stack.
- Because OpenVMS conditions can be caught in C++ catch handlers, the C++ RTL converts the conditions to an internal format that matches the representation of C++ exceptions. This conversion would sometimes lead to incorrect information being shown in the traceback. This problem has been fixed.

The following problems are fixed in this version of the C++ Library (Version 7.3 and higher compiler):

- As described in `http://issues.apache.org/jira/browse/STDCXX-397` (the Apache Software Foundation Issues website), the `__introsort_loop()` function in <algorithm.cc> header has a problem which, for some input sequences, can adversely affect performance of `std::sort`. For more

information, see the Apache tracker for the issue STDCXX-397 at http://issues.apache.org/jira/browse/STDCXX-397

The problem has been fixed. However, for some input sequences, the fix can change the behavior of std::sort with regard to the relative order in which elements that have equivalent ordering are placed into the sorted sequence. Though this change in behavior is permissible because, unlike std::stable_sort, std::sort does not guarantee any particular relative order of elements having equivalent ordering, to avoid breaking applications that rely on existing behaviour of std::sort, the fix is conditionalized with __RW_ FIX_APACHE_STDCXX_397 macro. The fix is in effect only when the program is compiled with this macro defined.

- When compiled in standard GNU mode, the library now defines the _RWSTD_ NO_IMPLICIT_INCLUSION macro, which causes library headers to include their respective template definition files. This is necessary because in standard GNU mode, implicit inclusion is disabled.

  Before this change, the program below would link with undefined symbol when compiled in standard GNU mode:

```
#include <vector>

int main() {
  std::vector<int> v;
  v.push_back(0);
}
```

- According to section 27.6.1.3 [lib.istream.unformatted] of the C++ Standard, the following get member functions of the std::basic_istream class should call setstate(failbit) if no characters have been stored, as is the case for an empty line. While on Integrity servers the functions set failbit, on Alpha systems they do not, for example:

```
istream_type& get(char_type *s, streamsize n, char_type delim);
istream_type& get(char_type *s, streamsize n);
```

## 5.3 Process/Application Hangs

The following restriction applies to the LIBRTL documentation for the lib$find_image_symbol run-time library routine:

If your application might dynamically activate shareable images that use pthreads (or the older CMA thread interface), the main image must be linked with the pthread$rtl image.

## 5.4 AST Delivery Clarification in Programs using POSIX Threads

*V8.3-1H1*

It is possible to utilize ASTs in threaded programs. Section B.12.5 in the Guide to the POSIX Threads Library describes some general usage notes and cautions. However, that section does not make clear how AST delivery behaves in programs with upcalls disabled (which is the default configuration).

In a program with upcalls disabled, user-mode ASTs will interrupt whatever thread happens to be executing at the moment that the AST is delivered. Therefore the AST service routine cannot make any assumptions about the context in which it executes (with respect to thread ID, stack space available, and so on.)

Also, note that much of the material in Section B.12.5 of the Guide describes a possible future version of OpenVMS. The description of generalized "per-thread" or thread-targeted ASTs represents possible future enhancements to the operating system. In all OpenVMS releases to date, however, user-mode ASTs are treated as if they are directed to the process as a whole.

## 5.5 RMS $PARSE Validation of Directory Files

*V8.3-1H1*

Starting with OpenVMS Version 8.3, the $PARSE service further validates all directories named in a directory specification to ensure that the directory characteristic is set. In previous OpenVMS versions, attempting to use a file with a .DIR extension that was not a directory resulted in a SS$_BADIRECTORY error from the $OPEN service, but not necessarily from the $PARSE service. As of Version 8.3, the error is consistently returned by the $PARSE service as long as it is not a syntax-only $PARSE.

## 5.6 No-IOLOCK8 Fibre Channel Port Drivers

*V8.3-1H1*

Many I/O subsystem components synchronize their operations across CPUs by using the IOLOCK8 spinlock, which has made acquiring the spinlock a performance bottleneck. As of Version 8.3-1H1, each Fibre Channel port driver (SYS$PGQDRIVER, SYS$PGADRIVER and SYS$FGEDRIVER) device uses its own port-specific spinlock instead of IOLOCK8 to synchronize its internal operations. In most configurations, this results in a significant decrease in the amount of time each CPU spends waiting for the IOLOCK8 spinlock as well as some increase in the Fibre Channel I/O rate.

Some minor changes are required to any class driver that connects to one of these new port drivers, so customers must determine whether they are running any non-HP class drivers that will not work with them. The simplest way to do this is to examine the output of the SDA command `CLUE SCSI/SUMMARY` and see whether the name of any third-party class driver device appears in the device hierarchy for an FGx0 or PGx0 port device in the "Device" column.

For more information, refer to the notes following this sample SDA session.

```
$ ANALYZE/SYSTEM

OpenVMS system analyzer

SDA> CLUE SCSI /SUMMARY
```

```
SCSI Summary Configuration:
---------------------------
SPDT      Port  STDT   SCSI-Id  SCDT  SCSI-Lun  Device    UCB       Type    Rev
--------------  -------------  --------------  --------  --------  ------  ----
81624200 FGB0
                8162CDC0     3
                                  8162D240     0  GGA22     8162F380  HSV200
                                  8162F180     1  DGA22801  8162FD40  HSV200  6100
                                  81632900     2  DGA22802  81632AC0  HSV200  6100
                                  816354C0     3  DGA22803  81635680  HSV200  6100
                                  81638080     4  DGA22804  81638240  HSV200  6100
                8162D400     4
                                  8162DD80     0  GGA22     8163AC40  MRD200
                                  8163B5C0     1  RJA22801  8163B780  RFD200  6100
                                  8163C840     2  RJA22802  8163CA00  RFD200  6100
                                  8163DAC0     3  RJA22803  8163DC80  RFD200  6100
                                  8163ED40     4  RJA22804  8163EF00  RFD200  6100
```

**Note**

All of the DGA and GGA devices in this output are accessed through the modified HP class drivers SYS$DKDRIVER and SYS$GKDRIVER respectively, so they are safe to use with the new port drivers.

Note that even though the physical device of Type MRD200 is not an HP qualified device, it does not present an IOLOCK8 problem because it is accessed through a GGAx unit, indicating that it uses the modified HP Generic class driver SYS$GKDRIVER.

The RJA devices are not controlled by a modified HP class driver so they will not work with the new port drivers.

## 5.7 C++ Compiler

*V8.3-1H1*

C++ Version 7.2 for OpenVMS for Integrity servers predefines the macro __INITIAL_POINTER_SIZE to 0, unlike C++ Version 7.1 compiler, which leaves it undefined. This is an intentional change that makes C++ Version 7.2 consistent with the C compiler and provides support for `pointer_size` pragmas, while C++ Version 7.1 does not.

Note that this change can cause diagnostics to appear in code that compiled cleanly with certain declarations selected by system header files that declare pointer types. This effect is most likely to appear in applications that use starlet headers and that compile with __NEW_STARLET defined.

If you cannot modify the application source code to conform to the new declarations, add the command-line qualifier /UNDEF=__INITIAL_POINTER_SIZE to the CXX command line to prevent the C++ Version 7.2 compiler from predefining this macro and thus causing the system headers to provide the same declarations as with Version 7.1 of the compiler.

## 5.8 Building DCE IDL C++ Applications

*V8.3-1H1*

Building DCE IDL C++ applications on CXX Version 7.2 and higher results in an undefined symbol linker warning. This is a known issue. To overcome this warning, contact HP Support Services to request any necessary patches.

## 5.9 Privileged Programs may Need a Recompile (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 may 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.

## 5.10 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 Integrity server 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 Integrity servers, 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.

### 5.10.1 KPB Extensions

*V8.2*

Prior versions of OpenVMS supported the use of KPBs for kernel mode above IPL 2. To make the transition to Integrity servers easier, usage of KPBs has been expanded for use in outer modes and all IPLs. This Alpha and Integrity servers change allows certain code that previously had private threading packages to make use of KPBs on both Alpha and Integrity servers. 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.

### 5.10.2 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 Integrity servers 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.

### 5.10.3 64-Bit Logical Block Number (LBN)

*V8.4*

OpenVMS today supports LBNs of only 32-bits. This limits our support of a disk volume to 2 TB. 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.

### 5.10.4 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.

### 5.10.5 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.

### 5.10.6 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.

### 5.10.7 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).

## 5.11 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 Integrity servers, 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 Integrity servers, refer to the OpenVMS Floating-Point White Paper on the following website:

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

### 5.11.1 IEEE Floating-Point Filter (Integrity servers 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.

### 5.11.2 Ada Event Support (Integrity servers 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 Integrity servers. 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.

### 5.11.3 C++ Language Issues (Integrity servers Only)

*V8.3*

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

**Workaround:** Compile C++ programs with /NOOPTIMIZE.

## 5.12 Ada Compiler(Integrity servers Only)

*V8.2*

GNAT Pro (Ada 83, Ada 95 and Ada 2005) is available from AdaCore. Contact AdaCore at www.adacore.com or sales@adacore.com for more information.

## 5.13 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.

## 5.14 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.

## 5.15 C Run-Time Library

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

### 5.15.1 C RTL TCP/IP Header File Updates

*V8.4*

The C RTL ships header files for users to call TCP/IP. C RTL places the headers into the C RTL header library (DECC$RTLDEF.TLB).

New header files are added, as appropriate for new features in TCP/IP.

```
SCTP.H
SCTP_UIO.H
```

These header files provide Stream Control Transmission Protocol (SCTP) support. For more information on SCTP, see the *HP TCP/IP Services for OpenVMS Version 5.7 Release Notes*.

### 5.15.2 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.

### 5.15.3 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.

### 5.15.4 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.

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

*V8.3*

The compare and swap built-ins (__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 built-ins visible.

### 5.15.6 Builtin __fci Added for Integrity servers

*V8.3*

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

### 5.15.7 No New Entries for DECC$*.OLB Object Libraries

*V8.3*

As of OpenVMS Alpha and Integrity servers 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.

## 5.16 Calling Standard and Rotating Registers (Integrity servers 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.

## 5.17 Common Data Security Architecture (CDSA) Considerations

The following notes pertain to CDSA.

### 5.17.1 Secure Delivery

*V8.4*

From Version 8.4 onwards, all OpenVMS kits including PCSI and VMSINSTAL based kits will be signed using HP Code Signing Service (HPCSS). The signing and validation do not use the CDSA infrastructure. A new companion file, <*full kit name*>_HPC is created and is provided along with the kit. The kit is then validated using this companion file.

A new validator, "HPBinarychecker" is installed on all OpenVMS systems to validate the kits signed using HP Code Signing Service. For more information, see *HP OpenVMS Version 8.4 New Features and Documentation Overview*.

Note that the kits that are signed prior to Version 8.4 can be validated using CDSA on OpenVMS Version 8.4.

*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.

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*.

### 5.17.2 Installation and Initialization Considerations

*V8.4*

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:

```
The following product has been selected:
    HP I64VMS CDSA T2.4-315               Layered Product

Do you want to continue? [YES]

%PCSI-E-HRDREF, product HP I64VMS CDSA T2.4-315 is referenced by HP I64VMS OPENV
MS X8.4-C42

    The two products listed above are tightly bound by a software dependency.
    If you override the recommendation to terminate the operation, the
    referenced product will be removed, but the referencing product will have
    an unsatisfied software dependency and may no longer function correctly.
    Please review the referencing product's documentation on requirements.

    Answer YES to the following question to terminate the PRODUCT command.
    However, if you are sure you want to remove the referenced product then
    answer NO to continue the operation.
```

## 5.18 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.

## 5.19 Delta/XDelta Debuggers

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

### 5.19.1 XDELTA Register Display Consideration (Integrity servers Only)

*V8.2*

XDELTA on OpenVMS Integrity server 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 5.16 for additional details.

## 5.20 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."

## 5.21 HP BLISS Compiler Warnings with RMS Structures (Integrity servers 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.

## 5.22 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.

## 5.23 HP COBOL Run-Time Library (RTL)

*V8.4*

For OpenVMS Alpha and OpenVMS Integrity servers Version 8.4, the HP COBOL RTL (DEC$COBRTL) has been updated to V2.9-785.

### 5.23.1 Performance Improvement of COBOL CALL Statement

*V8.4*

Performance degradation was observed on OpenVMS Integrity servers while calling subroutines, using the COBOL CALL statement. The time taken was much longer on OpenVMS Integrity servers than on OpenVMS Alpha.

The Run-Time Library (RTL) has been modified and the performance of the COBOL CALL statement has significantly improved. Now, the performance on OpenVMS Integrity servers is comparable to OpenVMS Alpha.

## 5.24 HP Fortran for Integrity servers

*V8.2*

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

## 5.25 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 Integrity servers. The following notes pertain to the MACRO compiler.

### 5.25.1 Enhancements for the Macro-32 Compiler

*V8.4*

The Macro-32 compiler has been enhanced to include several new instruction level built-ins for OpenVMS Alpha and OpenVMS Integrity server systems.

Table 5–1 summarizes the new built-ins added.

**Table 5–1 Macro-32 New Built-ins**

| Built-in | Operands[1] | Description | Supported on |
|---|---|---|---|
| EVAX_WH64 | <AB> | Generate Alpha write hint 64 instructions | Alpha |
| IA64_PADD1 | <WQ,RQ,RQ> | Generate Parallel Add instruction, single byte form with the first argument as the destination and the next two arguments as the source operands | Integrity servers |
| IA64_PADD1_SSS | <WQ,RQ,RQ> | Generate Parallel Add instruction, single byte form, with the first argument as the destination and the next two arguments as the source operands. Result and both source operands are treated as signed. | Integrity servers |
| IA64_PADD1_UUS | <WQ,RQ,RQ> | Generate Parallel Add instruction, single byte form, with the first argument as the destination and the next two arguments as the source operands. Result and first source operand are treated as unsigned and the second source operand is treated as signed. | Integrity servers |
| IA64_PADD1_UUU | <WQ,RQ,RQ> | Generate Parallel Add instruction, single byte form, with the first argument as the destination and the next two arguments as the source operands. Result and both source operands are treated as unsigned. | Integrity servers |

[1]The built-in requires the operands, where WQ - Write Quadword, PQ - Read Quadword, AB - Address of Byte.

(continued on next page)

**Table 5–1 (Cont.) Macro-32 New Built-ins**

| Built-in | Operands[1] | Description | Supported on |
|---|---|---|---|
| IA64_PADD2 | <WQ,RQ,RQ> | Generate Parallel Add instruction, two byte form, with the first argument as the destination and the next two arguments as the source operands. | Integrity servers |
| IA64_PADD2_SSS | <WQ,RQ,RQ> | Generate Parallel Add instruction, two byte form, with the first argument as the destination and the next two arguments as the source operands. Result and both source operands are treated as signed. | Integrity servers |
| IA64_PADD2_UUS | <WQ,RQ,RQ> | Generate Parallel Add instruction, two byte form, with the first argument as the destination and the next two arguments as the source operands. Result and first source operand are treated as unsigned and the second source operand is treated as signed. | Integrity servers |
| IA64_PADD2_UUU | <WQ,RQ,RQ> | Generate Parallel Add instruction, two byte form, with the first argument as the destination and the next two arguments as the source operands. Result and both source operands are treated as unsigned. | Integrity servers |
| IA64_PADD4 | <WQ,RQ,RQ> | Generate Parallel Add instruction, four byte form, with the first argument as the destination and the next two arguments as the source operands. | Integrity servers |
| IA64_MUX1 | <WQ,RQ,RQ> | Generate 'mux1' instruction with the first argument as the destination and the second argument as the source operand. The third argument specifies the permutation to be performed.[2] | Integrity servers |
| EVAX_S4ADDQ | <WQ,RQ,RQ> | Generates Alpha scaled quadword add instruction. | Alpha and Integrity servers |
| IA64_LFETCH_NT1, IA64_LFETCH_NT2, IA64_LFETCH_NTA, IA64_LFETCH_EXCL_NT1, IA64_LFETCH_EXCL_NT2, IA64_LFETCH_EXCL_NTA | <RQ,RQ> | These enhanced prefetch built-ins provide a method for specifying non-default cache locality hints (that is, ".nt1",".nt2",and ".nta"). The first operand is the address to prefetch and the second operand is for either the reg-base-update-form or the imm-baseupdate-form; if the operand is the literal zero, the no-baseupdate- form will be used. | Integrity servers |

[1]The built-in requires the operands, where WQ - Write Quadword, PQ - Read Quadword, AB - Address of Byte.

[2]A literal specifying the permutation has to be used as the third argument. The mapping of the permutations to the literals are as follows:

- BRCST 0
- MIX 8
- SHUF 9
- ALT 10
- REV 11

(continued on next page)

**Table 5–1 (Cont.) Macro-32 New Built-ins**

| Built-in | Operands[1] | Description | Supported on |
|---|---|---|---|
| All the _FAULT_variants of the LFETCH built-in | | Generates 'lfetch' instructions with the 'fault' completer. For example, IA64_LFETCH_FAULT_EXCL_NT1 | Integrity servers |

[1]The built-in requires the operands, where WQ - Write Quadword, PQ - Read Quadword, AB - Address of Byte.

For additional information about underlying instructions, see the respective hardware architecture manuals.

## 5.25.2 HP MACRO for OpenVMS Integrity servers

*V8.3*

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

- Prior to OpenVMS Version 8.3, the compiler generated the wrong code for the HALT instruction. On Integrity servers, 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 built-in 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 has to 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 built-in can be used as a workaround. See the following Macro-32 paragraphs for information about this built-in. 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 built-in can be used as a workaround. The IA64_LD8_A built-in, 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 built-in. The only workaround is to use /NOOPTIMIZE.

### 5.25.3 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
```

### 5.25.4 /OPTIMIZE=VAXREGS Qualifier Not Supported on Integrity servers

*V8.2*

The /OPTIMIZE=VAXREGS qualifier, which is supported on Alpha, is not supported on Integrity servers. Unfortunately, all of the related code was not removed from the command line processing. If you specify /OPTIMIZE=ALL on Integrity servers, 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.

### 5.25.5 Floating Divide-by-Zero Error Not Raised (Integrity servers 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.

# 5.26 Hypersort Utility

The following notes pertain to Hypersort V08-006 for OpenVMS Alpha and OpenVMS Integrity servers 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 5.38.

## 5.26.1 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.

## 5.26.2 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 5.26.8. If you encounter a hang or an ACCVIO with Hypersort, use SORT32 instead.

## 5.26.3 Hypersort and VFC Files Restriction

*V7.3-2*

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

## 5.26.4 /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.

### 5.26.5 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.

### 5.26.6 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.

### 5.26.7 Input Asterisk (*) Restriction

*Permanent Restriction*

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

### 5.26.8 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.

## 5.27 Intel Assembler (Integrity servers 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.

## 5.28 Librarian Utility

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

### 5.28.1 Linking Against Data-Reduced ELF Object Libraries Not Recommended (Integrity servers 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.

### 5.28.2 Failure to Insert or Replace .STB files in an Integrity servers Library (Integrity servers Only)

*V8.2*

On OpenVMS VAX and Alpha, .STB files can be stored in object libraries. On OpenVMS Integrity servers, 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 Integrity servers.

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 Integrity servers, 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.

### 5.28.3 Librarian Fails to Report Errors When Process Quota Too Low

*Permanent Restriction*

The OpenVMS Alpha and Integrity servers 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.

## 5.29 Linker Utility for OpenVMS Alpha

The following release notes pertain to the Alpha Linker Utility on all OpenVMS platforms. For Integrity servers Linker release notes, see Section 5.30.

# Programming Release Notes
# 5.29 Linker Utility for OpenVMS Alpha

## 5.29.1 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.

## 5.29.2 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`.

## 5.29.3 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.

# 5.30 Linker Utility for OpenVMS Integrity servers

The following release notes pertain to the Integrity servers Linker utility.

For Alpha Linker release notes, see Section 5.29.

## 5.30.1 Linker Writes Incorrect Interimage Debug Fixups into Debug Symbol File

*V8.3-1H1*

In some situations, the linker creates interimage fixups for the OpenVMS debugger. The inter-image 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 interimage fixups for the debugger. The C compiler rarely uses inter-image debugger fixups, although the C++ compiler often uses them. When linking such images with the /DEBUG qualifier, the linker writes the debug information followed by the interimage debug fixups. With the /NODSF qualifier (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 is corrected in the OpenVMS Version 8.3-1H1 Linker.

## 5.30.2 /SELECTIVE_SEARCH Qualifier Might Incorrectly Ignore Transfer Address

*V8.3-1H1*

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

In the following example, the object module (MAIN.OBJ) contains a transfer address but /SELECTIVE_SEARCH qualifier 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 Integrity server 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 the /SELECTIVE_SEARCH qualifier to the object module in the LINK command or in a LIBRARY command.

For example:

```
$ LINK MAIN/SELECTIVE_SEARCH
```

or

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

This problem manifests itself in one of two ways:

- The linker displays a warning message. 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. This 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 becomes the main entry point for the image. 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.

  This error is corrected in the OpenVMS Version 8.3-1H1 Linker.

### 5.30.3 Maximum Number of Sections

*V8.3-1H1*

For more than 65280 sections, the ELF format uses an extended numbering scheme, which was not implemented in the linker on OpenVMS Integrity server Version 8.3. As a result, the number of sections that a single input object module or shareable image can have was limited. Because the linker usually creates the shareable images with sections, this limit also applied when you created shareable images. In that case, ELF sections were used for exporting C++ templates or shared sections. That is, the maximum number of such interfaces in a shareable image had to be fewer than 65280.

In the OpenVMS Version 8.3-1H1 Linker, this restriction is removed. An input file, an object module or a shareable image can have more than 65280 sections.

### 5.30.4 Incorrect Creation Date of Shareable Images in the Map File

*V8.3-1H1*

On OpenVMS Integrity servers, shareable images sometimes showed an incorrect creation date in the linker map. The incorrect date shown was usually 3686. This condition occurred when the linker processed the shareable image as an input file and then extracted the date field, which was then shown in the map. The image itself had the correct date that you can see from ANALYZE/IMAGE output.

This error is corrected in the OpenVMS Version 8.3-1H1 Linker.

### 5.30.5 Demangler Information Look Up Results in Linker Access Violation

*V8.3-1H1*

In some cases, when the linker tried to look up demangler information in a shareable image that did not contain such information, the linker aborted with an access violation.

This problem is corrected in the OpenVMS V8.3-1H1 Linker.

### 5.30.6 Incorrect Secondary Messages for the NOGLOSYM Error Message

*V8.3-1H1*

For the NOGLOSYM error message, the OpenVMS Version 8.3 Linker printed the wrong relocation type and printed some information twice.

This problem is corrected in OpenVMS Version 8.3-1H1.

### 5.30.7 Incorrect Information for Undefined Symbols

*V8.3-1H1*

For undefined symbols, a USEUNDEF operation sometimes incorrectly showed information twice for the same reference. The problem occurred when the compiler generated a pair of relocations (LTOFF22X/LDXMOV) for the reference to an undefined symbol.

This problem is corrected in OpenVMS Version 8.3-1H1.

### 5.30.8 Incorrect UNMAPFIL Error

*V8.3-1H1*

If a non-ELF input file was encountered, the linker printed an INVLDHDR error message. After an INVLDHDR error, an incorrect UNMAPFIL error message was printed.

This problem is corrected in OpenVMS Version 8.3-1H1.

### 5.30.9 Max Ident Length Change for Shareable Images in Map

*V8.3-1H1*

In the linker map, the linker printed up to 14 characters of the ident from object modules and shareable images. (The maximum length of an ident for an object module is not limited; the maximum length of an ident for a shareable images is 15.)

The Version 8.3-1H1 linker now prints up to 15 characters as the maximum of the ident for a shareable image.

### 5.30.10 Linkage Type Check for Shareable Images

*V8.3-1H1*

Previously, the linker did not check the type and linkage for symbols from shareable images.

OpenVMS Version 8.3-1H1 Linker now performs this check.

### 5.30.11 Program Section Attribute ABS Ignored

*V8.3-1H1*

On Integrity servers, you cannot set the ABS attribute to a program section that contains labels to convert its offsets to constants. The linker prints the error message:

```
%ILINK-E-ILLABSPSC, absolute psect <psect-name> has non-zero length (not
allowed)
```

The OpenVMS 8.3-1H1 Linker ignores the ABS program section attribute and prints the following informational message:

```
%ILINK-I-PSCATTIGN, psect attribute ABS is not supported for OpenVMS ELF
sections, attribute ignored
```

### 5.30.12 Linker ACCVIOs when FP_MODE Literal Missing From Command Line

*V8.3-1H1*

In the Version 8.3, the Integrity server Linker encountered an access violation when the FP_MODE literal was missing on the command line.

This problem is corrected in OpenVMS Version 8.3-1H1.

### 5.30.13 OpenVMS Integrity servers Object Module and Image File Information Currently Unavailable

*V8.3*

Information about the OpenVMS Integrity servers extension to ELF (Executable and Linkable Format; the object module and image file format on Integrity servers) 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`

### 5.30.14 Differences Between the Integrity servers 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 Integrity servers 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 Integrity servers Linker.

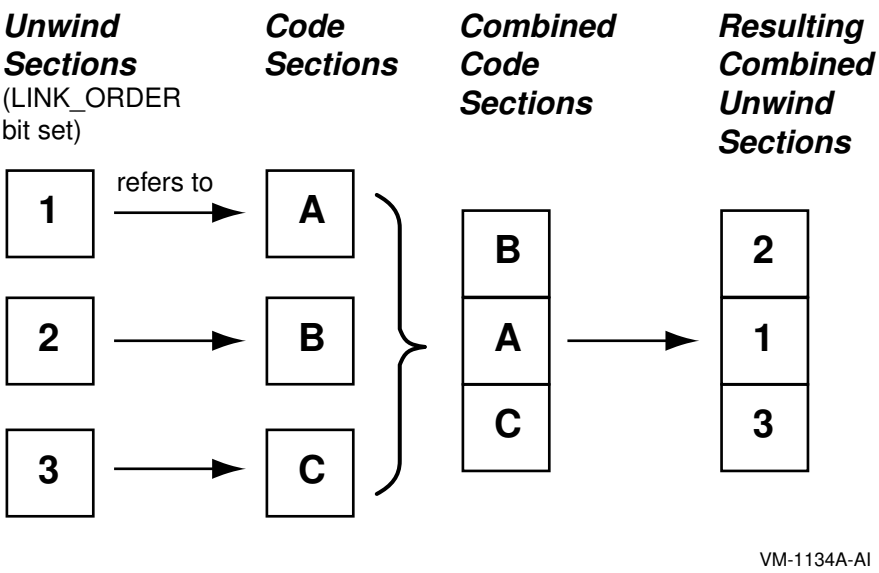
### 5.30.15 LINK_ORDER Section Header Flag Not Supported

*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.




### 5.30.16 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.

### 5.30.17 Error in Handling Initialized Overlaid Program Sections Fixed

*V8.3*

In previous versions of the Integrity servers linker, initialized overlaid program sections with zeros are incorrectly seen as compatible initializations. In OpenVMS Version 8.2 and Version 8.2-1, the Integrity servers 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.

### 5.30.18 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 Integrity servers 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 Integrity servers Linker in a future release of OpenVMS for Integrity servers.

### 5.30.19 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.

### 5.30.20 Better Use of Memory for Linker-Created Code Stubs

*V8.3*

The Integrity servers 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.

### 5.30.21 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.

## 5.31 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.

## 5.32 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.

## 5.33 OpenVMS System Dump Analyzer (SDA)

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

### 5.33.1 CLUE Commands Not Ported to OpenVMS Integrity servers

*V8.2*

The following CLUE commands have not yet been ported to OpenVMS Integrity servers and will return a message to that effect:

```
CLUE CALL
CLUE ERRLOG
CLUE FRU
CLUE REGISTER
```

## 5.34 PL/I Libraries Not Included in OpenVMS Integrity servers Version 8.2

*V8.2*

The PL/I libraries (native and translated) are not included in OpenVMS Integrity servers as they are in OpenVMS Alpha. The core PL/I images that are not available in OpenVMS Integrity servers 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 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 2.14.

## 5.35 POSIX Threads Library

The following sections contain release notes pertaining to the POSIX Threads Library (formerly named DECthreads).

# Programming Release Notes
# 5.35 POSIX Threads Library

## 5.35.1 Support for Process-shared Objects

*V8.2*

The POSIX Threads Library on OpenVMS Alpha and Integrity servers supports process-shared mutexes and condition variables.

**Note**

The process-shared read-write locks for mutexes and condition variables are supported only on Tru64 systems.

The supported pthread routines on OpenVMS are:

- pthread_condattr_getpshared
- pthread_condattr_setpshared
- pthread_mutexattr_getpshared
- pthread_mutexattr_setpshared

## 5.35.2 New Return Status for pthread_mutex_lock

*V8.2*

The POSIX Threads library pthread_mutex_lock when used with process-shared mutexes now returns a new error status known as EABANDONED. This status is returned only when the process has terminated while owning the mutex.

## 5.35.3 Support for New API pthread_mutex_tryforcedlock_np

*V8.2*

The POSIX Threads library supports a new API called pthread_mutex_ tryforcedlock_np. The pthread_mutex_tryforcedlock_np API takes the ownership of abandoned process-shared mutexes.

*Syntax*

```
pthread_mutex_tryforcedlock_np(mutex);
```

| Argument | Data Type | Access |
|---|---|---|
| mutex | opaque pthread_mutex_t | read |

*C Binding*

```
#include <pthread.h>
int
pthread_mutex_tryforcedlock_np (
          pthread_mutex_t *mutex);
```

*Argument*

```
mutex
Mutex to be locked.
```

*Description*

With process-shared objects, the objects such as mutexes and condition variables can be shared among multiple processes. If a process terminates while owning a mutex, the mutex will stay as owned by the terminated process. Hence none of the other threads or processes can own that mutex.

However, there is an optional system service `sys$pshared_register`. Usage of this system service makes the terminating process mark the mutexes that are owned by it as abandoned.

In such a scenario, the call to pthread_mutex_lock would return EABANDONED and the application could call pthread_mutex_tryforcedlock_np to get the ownership of the abandoned mutex.

*Return Values*

If an error condition occurs, this routine returns an integer value indicating the type of error. The possible return values can be:

| Return Value | Description |
|---|---|
| 0 | Successful completion |
| [EPERM] | The mutex is no longer abandoned, and is now owned by some other thread |
| [EINVAL] | The mutex is not a process-shared mutex |
| [ENOSYS] | Illegal system service called |

### 5.35.4 Stack Overflows During Exception Handling (Integrity servers Only)

*V8.2*

Exception handling on Integrity servers 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 Integrity servers. 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 Integrity servers. 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.

### 5.35.5 THREADCP Command Behavior on Integrity Servers

*V8.2*

The DCL command THREADCP cannot be used on OpenVMS Integrity servers 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 Integrity servers.

Alpha users should continue to use the THREADCP command.

### 5.35.6 Floating-Point Compilations and Exceptions (Integrity servers 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 Integrity servers 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 Integrity servers 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 Integrity servers. 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.

### 5.35.7 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.

### 5.35.8 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.

### 5.35.9 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.

### 5.35.10 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.

### 5.35.11 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.

## 5.36 RTL Library (LIB$)

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

### 5.36.1 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.

### 5.36.2 RTL Library (LIB$): Calling Standard Routines (Integrity servers 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.

## 5.37 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.

# 5.38 SORT32 Utility

The following release notes pertain to SORT32 V08-010 for OpenVMS Alpha and OpenVMS Integrity servers Version 8.2. For additional notes, refer to Section 5.26.8 and Section 5.26.1.

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 5.26.

## 5.38.1 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.

## 5.38.2 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.

## 5.38.3 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")))
```

## 5.38.4 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).

### 5.38.5 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.

## 5.39 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.

## 5.40 Watchpoint Utility (Integrity servers Only)

*V8.2*

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

## 5.41 Whole Program Floating-Point Mode (Integrity servers 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 Integrity servers, the floating-point rounding behavior, exception behavior, and precision control are defined at run time and are guided by the concept of whole program floating-point mode. With whole program floating-point mode, the module that contains the program main entry point (as determined by the Linker) is the module that defines the default floating-point rounding behavior, exception behavior, and precision control.

Most programs are not affected by this difference. Refer to the white paper ''OpenVMS Floating-Point Arithmetic on the Intel® Itanium® Architecture'' for essential reading. You can link to this document from the following web site:

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

# 6
# Hardware Release Notes

This chapter contains information about the following hardware products:

- USB Device Support (Section 6.1)
- MP and BMC Console (Section 6.2)
- AlphaServer 2100 (Section 6.3)
- AlphaServer 8200/8400 (Section 6.4)
- AlphaServer ES47/ES80/GS1280 Systems (Section 6.5)
- AlphaServer GS Series (Section 6.6)
- AlphaStation 200/400 (Section 6.7)
- AlphaStation 255 (Section 6.8)
- ATI RADEON 7000 Graphics (Section 6.9)
- ATI RADEON 7500 Graphics (Section 6.10)
- DECwindows X11 Display Server (Section 6.11)
- DIGITAL Modular Computing Components (Section 6.12)
- Digital Personal Workstation (Section 6.13)
- Dual-Controller HSG*nn* device (Section 6.14)
- Open3D Graphics (Section 6.15)
- PowerStorm 300/350 PCI Graphics Controller (Section 6.16)
- RF*nn* DSSI disk devices (Section 6.17)
- RZ*nn* disk devices (Section 6.18)
- sx1000 Integrity Superdome (Section 6.19)
- ZLX graphics boards (Section 6.20)

A few notes about using device drivers are also included at the end of this appendix.

## 6.1 USB Device Support

*V8.4*

OpenVMS tests and supports specific HP provided USB devices on HP Integrity servers.

In some cases, specific third-party devices are tested and supported by OpenVMS. For these devices, OpenVMS engineering will respond to bug reports and fix defects in driver support for them. These devices are listed in appendices of the *HP OpenVMS Systems Management Utilities Reference Manual*.

The nature of USB is such that in many cases broad categories of devices can be supported by a single generic device driver. OpenVMS does not attempt to prevent such unknown third-party devices from configuring and operating. However, we cannot guarantee correct operation of such untested devices. If a problem occurs with such a device, it can be reported through support channels but there is no guarantee that we will be able to correct the problem or provide an ECO patch for it.

Should you require formal support for a third-party device a request for support can be made through your HP account team or HP distributor.

## 6.2 MP and BMC Console Restrictions (Integrity servers Only)

The following notes pertain to the MP and BMC consoles.

### 6.2.1 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.

### 6.2.2 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.

## 6.3 AlphaServer 2100

The following sections contain information specific to the AlphaServer 2100 series computer.

### 6.3.1 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)
```

### 6.3.2 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.
```

## 6.4 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.

## 6.5 AlphaServer ES47/ES80/GS1280 Systems

This section contains release notes of interest to users of AlphaServer ES47/ES80/GS1280 systems. The note in Section 6.6.2 also applies to these systems.

### 6.5.1 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.

### 6.5.2 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*.

### 6.5.3 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.

### 6.5.4 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.

### 6.5.5 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.

## 6.6 AlphaServer GS Series Systems

This section contains release notes of general interest to most users of the AlphaServer GS Series systems.

### 6.6.1 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.

### 6.6.2 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.

### 6.6.3 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.

### 6.6.4 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.

## 6.7 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 |

## 6.8 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.

## 6.9 ATI RADEON 7000 Graphics (Integrity servers Only)

*V8.2*

The following release notes pertain to using the embedded ATI RADEON 7000 graphics adapter on OpenVMS Integrity servers.

Note: The resource requirements described in Section 6.10.1 also apply to the embedded ATI RADEON 7000 graphics adapter.

### 6.9.1 Integrity servers Graphics Support

*V8.2*

The Graphics option supported on OpenVMS Integrity servers are:

- ATI Radeon 7500 PCI
- ATI Radeon 7000 PCI (Embedded and Pluggable)
- ATI RN50 PCI

### 6.9.2 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.

## 6.10 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 6.15.

### 6.10.1 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.)

### 6.10.2 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.

### 6.10.3 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.

### 6.10.4 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.

### 6.10.5 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.

### 6.10.6 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
```

### 6.10.7 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.

### 6.10.8 Boot Reset Recommendation (Alpha Only)

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

### 6.10.9 No Overlay Planes

Hardware overlay planes are not supported.

### 6.10.10 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 6.10.16.)

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.

### 6.10.11 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.

### 6.10.12 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.

### 6.10.13 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.

### 6.10.14 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.

### 6.10.15 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.

### 6.10.16 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.

### 6.10.17 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
```

## 6.11 DECwindows X11 Display Server (Alpha Only)

This section contains release notes pertaining to the DECwindows X11 display server for OpenVMS Alpha systems.

### 6.11.1 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.

## 6.12 DIGITAL Modular Computing Components (DMCC)

This section contains release notes pertaining to DMCC.

### 6.12.1 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.

### 6.12.2 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
```

## 6.13 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.

## 6.14 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.

## 6.15 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 Integrity servers 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/`

## 6.16 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 6.15 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.

## 6.17 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 |
| 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.

## 6.18 RZ*nn* Disk Drive Considerations

The following notes describe issues related to various RZ disk drives.

### 6.18.1 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.

## 6.18.2 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.

## 6.18.3 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 6.18.3.1 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 6.18.3.2 to update the disk's firmware.

### 6.18.3.1 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<br>435 or 436 | RZ28_442D_DEC2104.FUP<br>RZ28P4_442C_DEC.FUP |

### 6.18.3.2 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.
```

## 6.19 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.

## 6.20 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)

## 6.21 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 5.10.

### 6.21.1 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 6.21.2.

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.

### 6.21.2 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 6.21.1.)

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*.

## 6.22 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.

## 6.23 Possible Per-Threads Security Impact on Alpha Device Drivers

*V7.2*

See Section 5.10.7 for information about how possible per-thread security impacts OpenVMS Alpha device drivers.

## 6.24 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.

## 6.25 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.

## 6.26 Adapter Release Notes

*V8.2-1*

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

### 6.26.1 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*.

### 6.26.2 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 Integrity servers 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

# 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.

## A.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.

## A.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.

## A.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 A.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 A.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.

## A.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 A.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.
```

## A.5 Compiler Versions

Table A–1 contains information about versions of compilers that might generate noncompliant code sequences and the recommended minimum versions to use when you recompile.

**Table A–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<br>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.

## A.6 Recompiling Code with ALONONPAGED_INLINE or LAL_REMOVE_FIRST

Any MACRO–32 code on OpenVMS Alpha that invokes either the ALONONPAGED_INLINE or the LAL_REMOVE_FIRST macro from the SYS$LIBRARY:LIB.MLB macro library must be recompiled on OpenVMS Version 7.2 or higher to obtain a correct version of these macros. The change to these macros corrects a potential synchronization problem that is more likely to be encountered on newer processors, starting with Alpha 21264 (EV6).

**Note**

Source modules that call the EXE$ALONONPAGED routine (or any of its variants) do *not* need to be recompiled. These modules transparently use the correct version of the routine that is included in this release.

# Index

## A


Ada compiler, 5–11
adapter release notes, 6–22
AlphaServer 2100
  console display, 6–3
  SCSI controller restriction, 6–3
AlphaServer 4100
  FRU table restriction, 6–4
AlphaServer 8200 systems
  FRU table restriction, 6–4
AlphaServer 8400 systems
  FRU table restriction, 6–4
AlphaServer ES47/ES80/GS1280 systems
  INIT console command usage on soft partitions, 6–4
  license requirement, 6–4
  RAD support, 6–4
  setting time at MBM, 6–5
  STOP/CPU and shutdown behavior, 6–4
AlphaServer GS Series systems
  device restriction, 6–5
  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, 4–34
API
  pthread_mutex_tryforcedlock_np, 5–34
Applications support for current release, 2–1
Associated products
  Software Public Rollout Reports, 2–1
  versions supported for current release, 2–1
AST delivery
  POSIX, 5–3
ATI RADEON 7000 graphics, 6–8
  hardware accelerated 3D graphics not supported, 6–9
ATI RADEON 7500 graphics, 6–9 to 6–13
  DECW$OPENGLSHR_RADEON.EXE renamed, 6–10
  DECwindows server hangs, 6–10
  OpenGL supports IEEE arithmetic only, 6–10
  video artifacts at high refresh rates, 6–10


## B


Backport library, 5–12
Backup API
  journaling events, 5–11
BL860c and BL870c servers, 4–5
BLISS
  See HP BLISS compiler
BMC console restrictions (Integrity servers only), 6–2
BUGCHECKFATAL system parameter, 5–15
<builtins.h> changes, 5–13


## C


C++ compiler
  See HP C++ compiler
C++ Run-time Library
  corrections, 5–1
CANCEL SELECTIVE function, improved use with LTDRIVER, 5–32
C compiler
  See HP C compiler
CDSA, 5–14
cell-based systems
  multiple nPartitions on, 4–30
Circuit switching
  and reduced performance, 4–28
CLUE commands
  not ported to Integrity servers, 5–33
Cluster compatibility kits, 4–26
Clusters
  See OpenVMS Cluster systems
CMAP files
  new, 2–4
COBOL RTL
  See HP COBOL RTL
COM for OpenVMS
  error with heavy load of applications, 2–4
  support, 2–4
Common Data Security Architecture
  See CDSA
Compilers
  noncompliant code, A–1, A–5

Configuring SAS tape drives, 4–12
C programs
  compiling and case sensitivity, 5–11
CPUSPINWAIT bugcheck, 5–15
CPU_POWER_MGMT default, 4–22
CRCTX routines enhanced, 6–22
C RTL, 5–11 to 5–13
  backport library, 5–12
  <builtins.h> changes, 5–13
  DECC$*.OLB libraries frozen, 5–13
  __fci built-in added, 5–13
  <time.h> changes, 5–12
C Run-Time Library
  See C RTL
Ctrl/H key sequence
  remapping to DEL (Integrity servers only), 6–2
Ctrl/P, 3–3

## D

Data-reduced ELF object libraries
  linking against, 5–31
DCE for OpenVMS, 2–2
DCL commands, 3–8
DDB structure
  updates, 5–8
DDT Intercept Establisher Routines
  device configuration, 4–28
DECC$*.OLB libraries frozen, 5–13
DECdfs for OpenVMS
  Version 2.4 required, 2–4
DECdtm
  Oracle 8*i*, 9*i*, 4–24
DECforms Web Connector
  running on OpenVMS Version 7.3-1 and higher, 2–5
DECmigrate
  not on V8.2 Open Source Tools CD, 3–9
DECnet/OSI
  See DECnet-Plus for OpenVMS
DECnet for OpenVMS, 1–3
DECnet-Plus for OpenVMS, 1–3
  new version required, 1–13
DEC PL/I, 2–5
DECram
  See HP DECram
DECRAM
  conflict with DECRYPT command, 2–6
DECwindows Motif
  See HP DECwindows Motif
DECwindows X11 display server, 6–13
  graphics boards support, 6–20
Delete key
  requires remapping (Integrity servers only), 6–2
Delta/XDelta debuggers, 5–15
  register display considerations, 5–15
Device configuration, 4–28
Device drivers
  IPL setup, 6–21
  MON version handling, 6–21
  per-thread security impact, 6–21
  recompiling and relinking, 6–20 to 6–21
  SCSI, 6–20
DEVICE_NAMING system parameter, 4–24
DIAGNOSE command
  not supported, 3–8
DIGITAL Modular Computing Components (DMCC)
  KZPDA controller and PBXGA graphics card, 6–13
  updating the SRM console, 6–13
Digital Personal Workstation, 6–14
Documentation changes and corrections
  *Guide to OpenVMS File Applications*, 5–16
  LIB$ help omission, 5–38
  *OpenVMS Performance Management*, 3–18
  *OpenVMS RTL Screen Management (SMG$) Manual*, 5–39
Documentation corrections, 3–10
  using IPC commands, 3–21
DSSI disk devices
  microcode revision levels, 6–16
Dual-controller HSG*nn*
  failure, 6–14
Dynamic CPU configuration
  POSIX Threads Library, 5–37

## E

EDIT/FDL
  fixing recommended bucket size, 4–24
EFI$CP utility
  use not recommended, 4–25
EFI driver, 6–22
ELV
  See Error Log Viewer (ELV) utility
Error Log Viewer (ELV) utility, 4–25
Error Message from ACPI, 4–10
EV6 Alpha processor, A–1
Extended DDT bit
  problem corrected, 5–32
Extended File Cache (XFC), 4–34
External authentication, 4–12
  Integrity servers support, 4–13
  password expiration notification, 4–13
  SET PASSWORD command, 4–13
External SAS disk device, 4–12

## F

F$GETSYI("RAD_CPUS"), 3–1
Fast Path
  turning off, for Galaxy on ES40, 4–31
__fci built-in added, 5–13
Fibre Channel, 5–4, 6–23
  compatibility kits, 4–26
  multipath failover restriction for tape devices, 4–29
Firmware
  for Alpha servers, 1–8
  for Integrity servers, 1–6
firmware, EFI, 4–5
Floating-point data
  considerations for applications, 5–10
FMS kits, 2–5
Fortran
  See HP Fortran
Freeware, 3–7
  menu unavailable on OpenVMS Integrity servers, 3–8

## G

Galaxy
  definitions, 4–30
Graphical Configuration Manager (GCM)
  supported for Galaxy, 4–30
Graphical Configuration Utility (GCU), 4–30
Graphics
  support for Integrity servers, 6–8
Graphics boards support, 6–20

## H

Hard partition, 4–30
HP BLISS compiler
  consequences of noncompliant code, A–1
  warnings (Integrity servers only), 5–16
HP C++ compiler
  consequences of noncompliant code, A–1
HP C compiler
  consequences of noncompliant code, A–1
HP COBOL RTL, 5–18
HP Code Signing Service for OpenVMS, 5–14
  HPCSS support, 3–1
HP DECram, 2–6
  command conflict between DECRAM and DECRYPT, 2–6
  removal before upgrade (Alpha only), 1–12
  ships as a SIP on V8.2, 2–6
  Version 2.5 (VAX only), 2–6
HP DECwindows Motif
  keyboard support restrictions (Integrity servers), 1–8
  user-written transport support, 2–7
HP DECwindows Motif (cont'd)
  version support, 1–4
HP Fortran
  for Integrity servers, 5–18
HP MACRO for OpenVMS, 5–19
  floating divide-by-zero error not raised (Integrity servers only), 5–22
  on Alpha systems, 5–22
  on Integrity servers, 5–21
  /OPTIMIZE=VAXREGS qualifier not supported on Integrity servers, 5–22
HP Secure Web Browser
  increased memory requirement, 3–9
  installation error on ODS-2 (Integrity servers only), 3–9
HP Secure Web Server
  support, 2–7
HP SIM, provisioning from, 4–5
HP SSL
  Startup commands for Encrypt and SSL, 1–12
HSG*nn*
  failure, 6–14
Hypersort utility, 5–23 to 5–24

## I

IDE CD, 6–14
ID-VSE for OpenVMS, 4–6
  simulated OpenVMS systems, 4–7
  sub-OS workloads, 4–7
  utilization data, 4–6
IEE Floating Point
  filter (Integrity servers only), 5–10
INIT console command
  usage on ES47/ES80/GS1280 soft partitions, 6–4
Insight software
  features not supported, 4–7
Installation and upgrade information
  networking options, 1–3
Installation error
  HP Secure Web Browser, 3–9
installing oracle database 10g Release 2, 3–4
INSTALL utility
  installing resident images, 4–35
Integrity servers
  firmware, 1–6
Integrity VM
  Guest Punishment Scenarios, 4–2
Intel Assembler (Integrity servers only), 5–24
Interlocked memory instructions, A–1
Invocation context block, 5–38
IPC Commands, 3–21

## K

Kerberos
  Kerberos for OpenVMS, 1–10
KPB extensions, 5–7

## L

LANCP
  converting device database after upgrading (Alpha only), 1–13
LAN Drivers
  duplex mode mismatch errors, 3–22
Large device name support, 4–10
Layered product
  fails to install, 1–14
LIB$GET_CURR_INVO_CONTEXT
  documentation correction, 3–20
LIB$GET_INVO_CONTEXT
  documentation correction, 3–20
LIB$GET_INVO_HANDLE
  documentation correction, 3–20
LIB$GET_PREV_INVO_CONTEXT
  documentation correction, 3–20
LIB$GET_PREV_INVO_HANDLE
  documentation correction, 3–20
LIB$GET_UIB_INFO
  documentation correction, 3–20
LIB$I64_GET_FR, 5–38
LIB$I64_GET_GR, 5–38
LIB$I64_PUT_INVO_REGISTERS, 5–38
LIB$I64_SET_FR, 5–38
LIB$I64_SET_GR, 5–38
LIB$LOCK_IMAGE
  missing from help, 5–38
LIB$PUT_INVO_REGISTERS
  documentation correction, 3–20
LIBRARIAN
  See Librarian Utility, 3–18
Librarian utility, 3–18, 5–24
  error reporting problem, 5–25
  linking against data-reduced ELF object libraries (Integrity servers restriction), 5–24
  restrictions with .STB files (Integrity servers only), 5–25
Library utility
  corrected information
    /accessing ELF object libraries, 3–19
    /REMOVE, 3–19
LIBRTL
  Calling Standard routines (Integrity servers only), 5–38
Licenses, 4–25
Licensing issues, 6–5 to 6–7
Linker for OpenVMS Alpha, 5–25 to 5–26
  change in behavior with library check, 5–26
  hangs when processing many files, 5–26
  limit of 25 elements on stack, 5–26
  RMS_RELATED_CONTEXT option, 5–26
Linker for OpenVMS Integrity servers, 5–27 to 5–32
  created code stubs, 5–32
  data-reduced ELF object libraries, 5–31
  demangled symbol names, 5–32
  differences from OpenVMS Alpha Linker, 5–30
  /EXPORT_SYMBOL_VECTOR removed, 5–32
  initialized overlaid program sections, 5–31
  LINK_ORDER section header flag, 5–30
  longer symbol names in options, 5–32
  /PUBLISH_GLOBAL_SYMBOLS removed, 5–32
LINK_ORDER ELF section header flag, 5–30
local area network, 4–5
Locales
  new, 2–7
LTDRIVER restriction, 5–32

## M

MACRO–32 compiler
  consequences of noncompliant code, A–1
  recompiling code, A–6
MACRO–64 assembler
  consequences of noncompliant code, A–1
MACRO for OpenVMS
  See HP MACRO for OpenVMS
Mail utility (MAIL)
  problem when callable mail used with kernel threads, 5–33
Microcode revision levels
  commands for updating, 6–18
  on DSSI disk devices, 6–16
Migration software, 1–14
MMG_CTLFLAGS system parameter, 3–18
Monitor utility changes, 4–19
MP console restrictions (Integrity servers only), 6–2
Multipath failover
  Fibre Channel tape device restriction, 4–29
  tape robots, 4–29
multiple nPartitions
  on cell-based systems, 4–30
multiple servers, provisioning, 4–5
MULTIPROCESSING system parameter, 5–15

## N

name length, InfoServer, 4–5
NetBeans
  Requires Java Standard Edition, Development Kit v 1.4.2-7, 2–2

Network
update restrictions, 3–21
Network options, 1–3
New Return Status
pthread_mutex_lock, 5–34
Noncompliant code, A–1, A–2

## O

Open3D graphics
controller boards support, 6–20
licensing change, 6–15
OpenVMS
ENCRYPT and DECRYPT commands, 1–12
OpenVMS Calling Standard
rotating registers, 5–13
OpenVMS Cluster systems, 4–14 to 4–29
compatibility kits, 4–26
compatibility kits for mixed versions, 4–26
patch kits, 4–25
performance reduced with CI-LAN switching, 4–28
rolling upgrades, 1–9
SCSI multipath failover, 4–27
OpenVMS Debugger
Ada event support, 5–10
C++ language issues, 5–11
OpenVMS DELTA/XDELTA Debugger
update to manual, 3–18
OpenVMS Galaxy, 4–29 to 4–31
and ES40
turning off Fast Path, 4–31
uncompressed dump limitation, 4–30
license enforcement, 6–5
OpenVMS Integrity servers
booting from DVD, 1–7
*OpenVMS Performance Management*
documentation correction, 3–18
OpenVMS Registry
Version 2 format database corruption, 4–31
OpenVMS System Dump Analyzer
CLUE commands not ported to Integrity servers, 5–33

## P

Parameters, 4–21
Partition
hard, 4–30
soft, 4–30
Pascal
reinstalling after an upgrade (Alpha), 2–8
V5.8A required to create STARLET library (Alpha only), 2–8
Patch kits
required for mixed-version OpenVMS Cluster system, 4–26
Patch kits for cluster compatibility, 4–25
PCB$T_TERMINAL
size increase, 5–8
PCI configuration restriction, 6–8
PEdriver
response to LAN congestion, 4–28
performance
COBOL CALL, 5–18
Performance, 4–9
Performance Enhancements, 4–8
Ctrl/T alignment faults, 4–10
dedicated CPU lock manager, 4–10
exception handling, 4–9
global section and creation and deletion, 4–10
image activation, 4–10
Per-thread security
impact on device drivers, 5–8
impact on privileged code, 5–8
PGFLQUOTA problems, 5–25
PL/I
libraries not included in Integrity servers, 5–33
RTL support, 2–5
POOLCHECK system parameter, 5–15
Port driver $QIO
restriction, 5–32
POSIX Threads Library, 5–33 to 5–38
debugger metering does not work, 5–38
dynamic CPU configuration, 5–37
floating-point exceptions (Integrity servers only), 5–36
pthread_mutex_lock, 5–34
pthread_mutex_tryforcedlock_np, 5–34
stack overflows during exception handling (Integrity servers only), 5–35
Support for Process-shared Objects, 5–34
THREADCP command behavior for Integrity servers, 5–36
PowerStorm 300/350 PCI graphics support, 6–15
Open3D license no longer checked, 6–15
Privileged data structures
64-bit logical block number, 5–7
CPU name space, 5–7
forking to a dynamic spinlock, 5–7
KPB extensions, 5–7
PCB$T_TERMINAL size increase, 5–8
per-thread security impact on, 5–8
UCB/DDB updates, 5–8
updates to, 5–6 to 5–8
provisioning
OpenVMS Guest limitation, 4–4
OpenVMS using HP SIM, 4–4
system firmware, 4–5
pthread_mutex_lock
New Return Status, 5–34
pthread_mutex_tryforcedlock_np
API, 5–34

## R

Recompiling programs
  for Alpha, 5–6
Remedial kits
  obtaining, 1–3
  required for mixed-version OpenVMS Cluster system, 4–26
restriction
  SYS$LDDRIVER, 4–22
RF73 and RF*nn* disks, controller memory errors, 6–16
RMS
  FAB, 5–17
Rotating registers, 5–38
Run-time library routines, 3–20
rx7620 server, 1–4
rx8620 server, 1–4
RZ*nn* disk drives, 6–18 to 6–20

## S

satellite system, 4–22
SCSI controllers
  restrictions on AlphaServer 2100 systems, 6–3
SCSI device drivers, 6–20
SCSI multipath incompatibility, 4–27
SDA
  See OpenVMS System Dump Analyzer
serial port enumeration, 3–4
Servers
  rx7620, 1–4
  rx8620, 1–4
  Superdome, 1–4
SET DEVICE/SWITCH command, 4–29
SET PASSWORD command, 4–13
SHUTDOWN.COM, 4–14
SMG$
  documentation corrections, 5–39
Software Public Rollout Reports, 2–1
SORT32 utility, 5–24, 5–40 to 5–41
SPLINVIPL bugcheck, 5–9
SRM_CHECK tool, A–1
storage controllers
  restriction, 1–4
Superdome
  sx1000, 6–20
Superdome server, 1–4
Support policy for software, 1–1
sx1000 chipset, 1–4
sx1000 Superdome, 6–20
symbolic debugger, 5–1
symlinks implementation, 3–1
SYS$GETTIM_PREC System Service, 3–1
SYS$SYSTEM:SHUTDOWN.COM command, 3–8
SYSGEN, 4–22
SYSMAN, 4–22
System crashes
  recovery from (Integrity servers only), 4–23
System disk
  incompatibility with older systems, 1–3
System Event Analyzer (SEA) utility
  support on Integrity servers, 2–8
System Event Log (SEL)
  clearing on Integrity servers, 1–5
System hangs
  recovery from (Integrity servers only), 4–23
System parameters, 4–31 to 4–32
  BUGCHECKFATAL, 5–15
  changes, 4–32
  DEVICE_NAMING
    used to increase device unit number maximum, 4–24
  MMG_CTLFLAGS documentation error, 3–18
  MULTIPROCESSING, 5–15
  new parameters, 4–31
  obsolete parameters, 4–31
  POOLCHECK, 5–15
  SYSTEM_CHECK, 5–15
System service changes, 5–1
SYSTEM_CHECK system parameter, 5–15

## T

Tape robots
  automatic multipath failover, 4–29
TCP/IP server components
  BIND, LPD, LBROKER, and SMTP, 4–5
TCP/IP Services for OpenVMS, 1–3
Terminal Fallback Facility (TFF), 4–32
  restrictions, 4–33
TFF
  See Terminal Fallback Facility
THREADCP command
  behavior for Integrity servers, 5–36
2 TiB disk volume support restrictions, 4–11
TIE kit, 1–14
<time.h> changes, 5–12
Timer queue entries (TQEs), 5–41
Time zone configuration, 4–1
TQEs
  See Timer queue entries
Translated Image Environment
  *See* TIE kit, 1–14
TZ function, 3–7, 4–22

## U

U160 SCSI Support
  rx7620, rx8620, 1–5

UCB structure
updates, 5–8
Upgrade
paths, 1–8
USB
device support, 6–1

## V

validation, 5–4
VAX Cluster Cache
See Virtual I/O Cache
VCC
See Virtual I/O Cache
VIOC
See Virtual I/O Cache
virtual connect, 4–23
failover, 4–23
Virtual I/O Cache
not available on Integrity servers, 4–34
superseded by XFC, 4–34
Volume Shadowing for OpenVMS
compatibility kits, 4–26

## W

Watchpoint utility, 5–41
WEBES
support on Integrity servers, 2–8
Whole-program floating-point mode (Integrity servers only), 5–41
Write Bitmaps, 4–8
write messages, 3–6

## X

XA, 4–24
XFC
See Extended File Cache

## Z

ZLX graphics boards support, 6–20