OpenVMS Version 7.3 New Features and Documentation Overview

Order Number: AA–QSBFD–TE

April 2001

This manual describes the new features associated with the OpenVMS Alpha and OpenVMS VAX Version 7.3 operating systems, and provides an overview of the documentation that supports the software.

Revision/Update Information: This is a new manual.

Software Version: OpenVMS Alpha Version 7.3
OpenVMS VAX Version 7.3

Compaq Computer Corporation
Houston, Texas

© 2001 Compaq Computer Corporation

Compaq, AlphaServer, POLYCENTER, VAX, VMS, and the Compaq logo Registered in U.S. Patent and Trademark Office.

OpenVMS and Tru64 are trademarks of Compaq Information Technologies Group, L.P. in the United States and other countries.

UNIX and X/Open are trademarks of The Open Group in the United States and other countries.

All other product names mentioned herein may be trademarks of their respective companies.

Confidential computer software. Valid license from Compaq 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.

Compaq shall not be liable for technical or editorial errors or omissions contained herein. The information in this document is provided "as is" without warranty of any kind and is subject to change without notice. The warranties for Compaq products are set forth in the express limited warranty statements accompanying such products. Nothing herein should be construed as constituting an additional warranty.

ZK6620

The Compaq *OpenVMS* documentation set is available on CD-ROM.

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

Contents

Preface ix

Part I OpenVMS Version 7.3 New Features

1 Summary of OpenVMS Version 7.3 New Features

2 Compaq OpenVMS e-Business Features

2.1 Compaq Secure Web Server for OpenVMS Alpha 2–2
2.2 Compaq COM for OpenVMS 2–2
2.3 Compaq Java 2 SDK, Standard Edition, for OpenVMS Alpha 2–2
2.4 Compaq Fast Virtual Machine (Fast VM) for the Java 2 Platform 2–2
2.5 Compaq XML Technology 2–2
2.6 Attunity Connect "On Platform" Package 2–3
2.7 Compaq Enterprise Directory Services for e-Business 2–3
2.8 Reliable Transaction Router (RTR) 2–3
2.9 Compaq BridgeWorks 2–3

3 General User Features

3.1 DCL Commands and Lexical Functions 3–1
3.2 New Online Help Topics for Utility Routines 3–3
3.3 MIME Utility Enhancements 3–3
3.4 WWPPS Utility (Alpha) 3–4

4 System Management Features

4.1 OpenVMS Support for New AlphaServer GS Series Systems 4–1
4.1.1 OpenVMS Support for Hard and Soft Partitions 4–1
4.1.2 OpenVMS Resource Affinity Domain (RAD) Support for Applications 4–1
4.1.3 OpenVMS Support for CPU Online Replace 4–2
4.2 Automatic Setting of Daylight Savings Time 4–2
4.3 Class Scheduler for CPU Scheduling 4–2
4.4 Dedicated CPU Lock Manager (Alpha) 4–4
4.4.1 Implementing the Dedicated CPU Lock Manager 4–4
4.4.2 Enabling the Dedicated CPU Lock Manager 4–4
4.4.3 Using the Dedicated CPU Lock Manager With Affinity 4–5
4.4.4 Using the Dedicated CPU Lock Manager with Fast Path Devices 4–5
4.4.5 Using the Dedicated CPU Lock Manager on the AlphaServer GS Series Systems 4–6
4.5 OpenVMS Enterprise Directory for e-Business (Alpha) 4–6
4.6 Extended File Cache (Alpha) 4–7

4.7 /ARB_SUPPORT Qualifier Added to INSTALL Utility (Alpha) 4–7
4.8 MONITOR Utility New Features 4–7
4.9 OpenVMS Cluster Systems 4–8
4.9.1 Clusterwide Intrusion Detection 4–8
4.9.2 Fast Path for SCSI and Fibre Channel (Alpha) 4–9
4.9.3 Floppy Disks Served in an OpenVMS Cluster System (Alpha) 4–9
4.9.4 New Fibre Channel Support (Alpha) 4–9
4.9.4.1 New Fibre Channel Tape Support (Alpha) 4–10
4.9.5 LANs as Cluster Interconnects 4–10
4.9.5.1 SCA Control Program 4–11
4.9.5.2 New Error Message About Packet Loss 4–11
4.9.6 Warranted and Migration Support 4–11
4.10 OpenVMS SMP Performance Improvements (Alpha) 4–12
4.11 New SYSMAN Commands and Qualifiers 4–14
4.12 New System Parameters 4–15
4.12.1 AUTO_DLIGHT_SAV 4–15
4.12.2 FAST_PATH_PORTS 4–15
4.12.3 GLX_SHM_REG 4–15
4.12.4 LCKMGR_CPUID (Alpha) 4–15
4.12.5 LCKMGR_MODE (Alpha) 4–16
4.12.6 NPAGECALC 4–16
4.12.7 NPAGERAD (Alpha) 4–16
4.12.8 RAD_SUPPORT (Alpha) 4–17
4.12.9 SHADOW_MAX_UNIT 4–17
4.12.10 VCC_MAX_IO_SIZE (Alpha) 4–17
4.12.11 VCC_READAHEAD (Alpha) 4–17
4.12.12 WBM_MSG_INT 4–18
4.12.13 WBM_MSG_LOWER 4–18
4.12.14 WBM_MSG_UPPER 4–19
4.12.15 WBM_OPCOM_LVL 4–19
4.13 Volume Shadowing for OpenVMS 4–19
4.13.1 Minicopy in Compaq Volume Shadowing for OpenVMS (Alpha) 4–20
4.13.2 New Volume Shadowing Features for Multiple-Site OpenVMS Cluster Systems 4–21
4.13.2.1 How to Use the New DISMOUNT and SET Command Qualifiers 4–24
4.13.3 Using INITIALIZE/SHADOW/ERASE to Streamline the Formation of a Shadow Set 4–25

5 Programming Features

5.1 3D Graphics Support 5–1
5.2 3X-DAPBA-FA and 3X-DAPCA-FA ATM LAN Adapters (Alpha) 5–2
5.3 Compaq COBOL Run-Time Library Enhancements 5–2
5.4 Compaq C Run-Time Library Enhancements 5–2
5.4.1 Strptime Function Is XPG5-Compliant 5–2
5.4.2 Limitation of Eight Nested Directory Levels Was Lifted (Alpha) 5–3
5.4.3 Improved Support for Extended File Specifications (Alpha) 5–3
5.4.3.1 Compaq C RTL Supports Case Preservation in File Names 5–3
5.4.3.2 Most C RTL Functions Now Accept Long OpenVMS File Names As Arguments (Alpha) 5–3
5.4.4 Compaq C RTL Supports Exact Case Argv Arguments (Alpha) 5–4
5.4.5 Compaq C RTL Can Implicitly Open Files for Shared Access 5–4
5.4.6 Alternate Way of Translating UNIX File Specifications 5–5

5.4.7 New Functions 5–5
5.5 Fortran Support for 64-Bit Address (Alpha) 5–5
5.6 Large Page-File Sections (Alpha) 5–5
5.7 Multipath System Services 5–6
5.8 Multiprocess Debugging (Alpha) 5–7
5.9 Performance Application Programming Interface (API) 5–8
5.10 POLYCENTER Software Installation Utility Enhancements 5–8
5.11 New Process Dump Tools (Alpha) 5–8
5.11.1 DCL ANALYZE/PROCESS_DUMP Command 5–9
5.11.2 Debugger ANALYZE/PROCESS_DUMP Command 5–9
5.11.3 Debugger SDA Command 5–9
5.11.4 Analyzing Process Dumps on Different Systems 5–10
5.11.5 Forcing a Process Dump 5–10
5.11.6 Process Dumps: Security and Diskquota Guidelines 5–10
5.11.6.1 Special Rights Identifiers 5–11
5.11.6.2 Privileged Users and Process Dumps 5–11
5.11.6.3 Nonprivileged Users and Process Dumps 5–11
5.11.6.4 Protecting Process Dumps 5–11
5.12 RMS Locking Enhancements 5–12
5.12.1 RMS Locking Performance Enhancements (Alpha) 5–12
5.12.1.1 RMS Global Buffer Read-Mode Locking 5–12
5.12.1.2 No Query Record Locking Option 5–13
5.12.2 Record Locking Options to Control Deadlock Detection 5–15
5.13 OpenVMS Registry 5–16
5.13.1 REG$CP Registry Utility 5–16
5.14 Alpha SDA Commands, Parameters, and Qualifiers 5–17
5.14.1 New Alpha SDA Commands 5–17
5.14.1.1 DUMP 5–17
5.14.1.2 SET SYMBOLIZE 5–18
5.14.1.3 SHOW MEMORY 5–19
5.14.1.4 SHOW RAD 5–19
5.14.1.5 SHOW TQE 5–20
5.14.1.6 UNDEFINE 5–20
5.14.2 New Parameters and Qualifiers for Existing Commands 5–20
5.14.2.1 REPEAT 5–21
5.14.2.2 SEARCH 5–21
5.14.2.3 SET OUTPUT 5–21
5.14.2.4 SET PROCESS 5–21
5.14.2.5 SHOW DEVICE 5–22
5.14.2.6 SHOW GCT 5–22
5.14.2.7 SHOW LOCK 5–22
5.14.2.8 SHOW PFN_DATA 5–22
5.14.2.9 SHOW POOL 5–22
5.14.2.10 SHOW PROCESS 5–23
5.14.2.11 SHOW RESOURCE 5–24
5.14.2.12 SHOW SPINLOCKS 5–24
5.14.2.13 SHOW SUMMARY 5–24
5.15 New SDA Commands for the Spinlock Tracing Utility 5–24
5.15.1 SPL LOAD 5–24
5.15.2 SPL SHOW COLLECT 5–24
5.15.3 SPL SHOW TRACE 5–25
5.15.4 SPL START COLLECT 5–25
5.15.5 SPL START TRACE 5–26
5.15.6 SPL STOP COLLECT 5–26

5.15.7 SPL STOP TRACE 5–26
5.15.8 SPL UNLOAD 5–26
5.16 System Services 5–27
5.17 TCP/IP Files for SDA READ Command That Contain Global Symbols and Locations 5–28
5.18 Visual Threads Version 2.1 (Alpha) 5–29

6 Associated Products Features

6.1 Availability Manager 6–1
6.2 Compaq Advanced Server V7.3 for OpenVMS (Alpha) 6–2
6.3 Compaq DECwindows Motif for OpenVMS 6–2
6.4 Compaq DCE for OpenVMS 6–3
6.4.1 Compaq DCE Remote Procedure Call (RPC) 6–3
6.4.2 New Ethernet Device Support 6–3
6.4.3 For More DCE Information 6–3
6.5 DECram for OpenVMS Version 3.0 (Alpha) 6–3
6.6 Enterprise Capacity and Performance (ECP) 6–4
6.6.1 ECP Collector for OpenVMS Version 5.4 6–4
6.6.2 ECP Performance Analyzer for OpenVMS Version 5.4 6–4
6.7 Kerberos for OpenVMS 6–5
6.7.1 New DCL Command KERBEROS 6–5
6.8 Universal LDAPv3 API (Alpha) 6–5
6.9 Compaq PATHWORKS V6.0D for OpenVMS (Advanced Server) 6–6
6.10 Compaq Service Tools and DECevent 6–6
6.11 Compaq TCP/IP Services for OpenVMS Version 5.1 6–7
6.11.1 New Features and Changes 6–7
6.11.2 TCP/IP Services for OpenVMS Documentation 6–8

Part II Overview of OpenVMS Documentation

7 OpenVMS Documentation Overview

7.1 OpenVMS Documentation Changes 7–1

8 OpenVMS Printed and Online Documentation

8.1 Printed Documentation 8–1
8.1.1 OpenVMS Media Kit Documentation 8–1
8.1.2 OpenVMS Documentation Sets 8–2
8.1.3 Documentation for System Integrated Products 8–4
8.1.4 Archived OpenVMS Documentation 8–4
8.2 Online Documentation on CD-ROM 8–4
8.2.1 Online Formats 8–4
8.2.2 PDF Viewers 8–5
8.3 Online Documentation on the OpenVMS Web Site 8–5
8.4 Online Help 8–5

9 Descriptions of OpenVMS Manuals

9.1 Manuals Included in the OpenVMS Media Kit 9–1
9.2 Manuals in the OpenVMS Documentation Set 9–1
9.2.1 OpenVMS Base Documentation Set 9–1
9.2.2 Continuation of Full Documentation Set 9–2
9.3 OpenVMS Alpha Device Driver Manual 9–7
9.4 RMS Journaling Manual 9–7
9.5 Archived Manuals 9–7

Index

Figures

4–1 Application Writes to a Write Bitmap 4–20
4–2 Member Returned to the Shadow Set (Virtual Unit) 4–20

Tables

1–1 Summary of OpenVMS VAX and OpenVMS Alpha Version 7.3 Software Features 1–3
4–1 SYSMAN command: class_schedule 4–3
4–2 OpenVMS Cluster Warranted and Migration Support 4–12
5–1 PDL Changes 5–8
5–2 Methods Available for Specifying No Query Record Locking 5–14
5–3 Qualifiers for the SPL SHOW TRACE Command 5–25
5–4 Qualifiers for the SPL START COLLECT Command 5–25
5–5 Qualifiers for the SPL START TRACE Command 5–26
5–6 Modules Containing Global Symbols and Data Structures Used by SDA 5–28
5–7 Modules Defining Global Locations Within the Executive Image 5–29
7–1 Changes to Documentation for OpenVMS Version 7.3 7–1
8–1 OpenVMS Media Kit Manuals 8–1
8–2 OpenVMS Full Documentation Set (QA-001AA-GZ.7.3) 8–2
9–1 Archived OpenVMS Manuals 9–7
9–2 Archived Networking Manuals and Installation Supplements 9–9

Preface

Intended Audience

This manual is intended for general users, system managers, and programmers who use the Compaq *OpenVMS* Operating System.

This document describes the new features related to Version 7.3 of the OpenVMS Alpha and OpenVMS VAX operating systems. For information about how some of the new features might affect your system, read the release notes before you install, upgrade, or use Version 7.3.

Document Structure

This manual contains the following parts and chapters:

- Part I, OpenVMS Version 7.3 New Features
  - Chapter 1 contains a summary of the new OpenVMS software features.
  - Chapter 2 provides information on the e-Business technologies that are included in the Compaq OpenVMS e-Business Infrastructure Package with OpenVMS Alpha Version 7.3.
  - Chapter 3 describes new features of interest to general users of the OpenVMS VAX and OpenVMS Alpha operating systems.
  - Chapter 4 describes new features that are applicable to the tasks performed by system managers.
  - Chapter 5 describes new features that support programming tasks.
  - Chapter 6 describes significant layered product new features.
- Part II, Overview of OpenVMS Documentation
  - Chapter 7 describes the OpenVMS documentation changes from the previous version.
  - Chapter 8 describes how the documentation is delivered.
  - Chapter 9 describes each manual in the OpenVMS documentation set.

Related Documents

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

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

Reader's Comments

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

Internet	**openvmsdoc@compaq.com**
Mail	Compaq Computer Corporation OSSG Documentation Group, ZKO3-4/U08 110 Spit Brook Rd. Nashua, NH 03062-2698

How To Order Additional Documentation

Use the following World Wide Web address to order additional documentation:

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

If you need help deciding which documentation best meets your needs, call 800-282-6672.

Conventions

The following conventions are used in this manual:

Ctrl/*x*	A sequence such as Ctrl/*x* indicates that you must hold down the key labeled Ctrl while you press another key or a pointing device button.
PF1 *x*	A sequence such as PF1 *x* indicates that you must first press and release the key labeled PF1 and then press and release another key or a pointing device button.
Return	In examples, a key name enclosed in a box indicates that you press a key on the keyboard. (In text, a key name is not enclosed in a box.) In the HTML version of this document, this convention appears as brackets, rather than a box.
. . .	A horizontal ellipsis in examples indicates one of the following possibilities: • Additional optional arguments in a statement have been omitted. • The preceding item or items can be repeated one or more times. • Additional parameters, values, or other information can be entered.
. . .	A vertical ellipsis indicates the omission of items from a code example or command format; the items are omitted because they are not important to the topic being discussed.
( )	In command format descriptions, parentheses indicate that you must enclose the options in parentheses if you choose more than one.
[ ]	In command format descriptions, brackets indicate optional elements. You can choose one, none, or all of the options. (Brackets are not optional, however, in the syntax of a directory name in an OpenVMS file specification or in the syntax of a substring specification in an assignment statement.)

[|]	In command format descriptions, vertical bars separating items inside brackets indicate that you choose one, none, or more than one of the options.
{ }	In command format descriptions, braces indicate required elements; you must choose one of the options listed.
bold text	This text style represents the introduction of a new term or the name of an argument, an attribute, or a reason. In the HTML version of this Conventions table, this convention appears as *italic text*.
italic text	Italic text indicates important information, complete titles of manuals, or variables. Variables include information that varies in system output (Internal error *number*), in command lines (/PRODUCER=*name*), and in command parameters in text (where *dd* represents the predefined code for the device type).
UPPERCASE TEXT	Uppercase text indicates a command, the name of a routine, the name of a file, or the abbreviation for a system privilege.
`Monospace text`	Monospace type indicates code examples and interactive screen displays. In the C programming language, monospace type in text identifies the following elements: keywords, the names of independently compiled external functions and files, syntax summaries, and references to variables or identifiers introduced in an example. In the HTML version of this Conventions table, this convention appears as *italic text*.
-	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.

Part I

OpenVMS Version 7.3 New Features

1

Summary of OpenVMS Version 7.3 New Features

Compaq OpenVMS Version 7.3 delivers the highest levels of availability, scalability, flexibility, performance, and security that are required for e-Business. With more than 20 years of proven reliability, OpenVMS continues to improve its availability and performance by including new technology into the base operating system and the OpenVMS Cluster software environment. The base infrastructure technologies needed to support e-Business applications are being incorporated into the OpenVMS license, so that OpenVMS is e-Business and Internet ready.

OpenVMS Version 7.3 new features include the following:

- Continuous improvement of OpenVMS availability
  - Improved disaster tolerant capabilities with ATM & Gigabit Ethernet as cluster interconnects because of their inherent high performance and long distance capabilities.
  - Greater system availability during backups with faster reintegration of shadow members after copies.
  - Enhanced OpenVMS support for the latest ENSA Storage offerings.
  - Greater serviceability for the new AlphaServer GS systems with CPU Online Replace.
- Higher system and application performance
  - Substantial application performance benefits can be realized with the new Extended File Cache.
  - Improved cluster, SMP and system performance through enhancements to RMS and the clusterwide Lock Manager, as well as FastPath support for Fibre Channel and SCSI storage architectures.
- Expanded OpenVMS security options
  - OpenVMS adds industry standard Kerberos Version 5 security, which allows OpenVMS applications to communicate and authenticate with Windows 2000 domains and UNIX platforms.
  - OpenVMS Intrusion Detection expands to be clusterwide, reducing chances of system break-ins.
- Expanded OpenVMS license includes new e-Business technologies
  - Compaq Secure Web Server for OpenVMS Alpha (based on Apache)
  - Compaq COM for OpenVMS
  - Compaq BridgeWorks
  - Extensible Markup Language API (XML parser)

- Java 2 SDK, Standard Edition
- Attunity Connect "On Platform Package" (native JDBC driver included)
- Reliable Transaction Router (RTR)
- OpenVMS Enterprise Directory for e-Business (X.500)

OpenVMS Version 7.3 includes all the capabilities of OpenVMS Version 7.2, OpenVMS Version 7.2-1, and the OpenVMS Version 7.2-1H1 hardware release.

Table 1–1 summarizes each feature provided by OpenVMS Alpha and OpenVMS VAX Version 7.3 and presents these features according to their functional component (e-Business, general user, system management, programming, or associated product).

New features that are available only on the OpenVMS Alpha platform are identified with the word Alpha in the summary table and in the section title. These features are not available on OpenVMS VAX systems. However, some of the OpenVMS Alpha features may be part of a mixed architecture cluster.

Table 1–1 Summary of OpenVMS VAX and OpenVMS Alpha Version 7.3 Software Features

Compaq OpenVMS e-Business Infrastructure Package for OpenVMS Alpha Version 7.3	
Compaq OpenVMS e-Business	This chapter provides information on the following e-Business technologies that are included in the Compaq OpenVMS e-Business Infrastructure Package with OpenVMS Alpha Version 7.3: • Compaq Secure Web Server for OpenVMS Alpha (based on Apache) • Compaq COM for OpenVMS • Compaq Java 2 SDK, Standard Edition • Compaq Fast Virtual Machine (Fast VM) for the Java 2 Platform on OpenVMS Alpha • Compaq XML Technology • Attunity Connect "On Platform" Package • Compaq Enterprise Directory Services for e-Business • Reliable Transaction Router (RTR) • Compaq Bridgeworks The Compaq OpenVMS e-Business Infrastructure Package provides key Internet and e-Business software technology that enhances the base OpenVMS Alpha operating system. These technologies are licensed with the OpenVMS Alpha operating system.
General User Features	
DCL commands	Several DCL commands have been updated for Version 7.3.
Utility routines online help	As of Version 7.3, online help now includes all the OpenVMS utility routines that are described in the *OpenVMS Utility Routines Manual*.
MIME utility	The MIME utility has been enhanced with new commands and qualifiers.
OpenVMS Alpha firmware	The OpenVMS Version 7.3 CD-ROM package includes the Alpha Systems Firmware Update Version 5.9 CD-ROM and Release Notes. Please read the Release Notes before installing the firmware.
World-Wide PostScript Printing Subsystem (Alpha)	WWPPS is a new utility that allows users to print text files in many languages on generic PostScript printers that do not have resident language fonts.
System Management Features	
OpenVMS support for new AlphaServer GS series systems	OpenVMS Version 7.3 provides support for Compaq's AlphaServer GS80, GS160 and GS320 systems. This support includes: • OpenVMS support for hard and soft partitions (Galaxy) on AlphaServer GS160 and GS320 systems • OpenVMS Resource Affinity Domain (RAD) support for applications • CPU Online Replace
Automatic daylight savings time change	System parameter can be set to allow OpenVMS to automatically change system time to and from Daylight Savings Time.

(continued on next page)

Table 1–1 (Cont.) Summary of OpenVMS VAX and OpenVMS Alpha Version 7.3 Software Features

System Management Features	
CPU Online Replace (AlphaServer GS160/320 systems)	OpenVMS Alpha Version 7.3 supports CPU Online Replace. CPU Online Replace provides the ability to replace secondary CPUs on a running system without rebooting, which increases system maintainability and serviceability.
Class scheduler for CPU scheduling	A new class scheduler for both VAX and Alpha systems allows you to designate the amount of CPU time that a system's users may receive by placing users into scheduling classes.
Dedicated CPU Lock Manager (Alpha)	Compaq for OpenVMS Version 7.3 provides a dedicated CPU lock manager, which improves SMP system and application performance.
OpenVMS Enterprise Directory for e-Business	OpenVMS Enterprise Directory for e-Business is a massively scalable directory service, providing both X.500 and LDAPv3 services on OpenVMS Alpha with no separate license fee.
Extended File Cache (XFC) (Alpha)	XFC improves I/O performance and gives you control over the choice of cache and cache parameters. The Extended File Cache improves I/O performance with the following features that are not available with the Virtual I/O Cache: • Read-ahead caching • Automatic resizing of the cache • Larger maximum cache size • No limit on the number of closed files that can be cached • Control over the maximum size of I/O that can be cached • Control over whether cache memory is static or dynamic
INSTALL utility has /ARB_ SUPPORT qualifier	You can use the /ARB_SUPPORT qualifier with the ADD, CREATE, and REPLACE commands in the INSTALL Utility. The ARB_SUPPORT qualifier provides Access Rights Block (ARB) support to products that have not yet been updated with the per-thread security Persona Security Block (PSB) data structure.
MONITOR utility enhancements	Two new MONITOR command parameters have been added to enhance the OpenVMS display of system information. RLOCK and TIMER are used to monitor, respectively, the dynamic lock remastering statistics and the Timer Queue Entry (TQE) statistics of a node.

(continued on next page)

Table 1–1 (Cont.) Summary of OpenVMS VAX and OpenVMS Alpha Version 7.3 Software Features

System Management Features		
OpenVMS Cluster systems	The following are new OpenVMS Cluster features:	
	Clusterwide intrusion detection	Clusterwide intrusion detection is enabled by the integration of intrusion data and information from each system into a clusterwide intrusion database. The clusterwide database contains all unauthorized attempts and the state of any intrusion event.
	Fast Path for SCSI and Fibre Channel (Alpha)	Fast Path is now supported on KZPBA (parallel SCSI) and KGPSA (Fibre Channel) ports.
	Floppy disk served in an OpenVMS Cluster system (Alpha)	MSCP supports the serving of floppy disks in an OpenVMS Cluster system, provided the devices are named in accordance with the port allocation naming conventions.
	Fibre Channel support (Alpha)	Support for larger I/O packets, for new Fibre Channel hardware, and for larger configurations is included.
	LANs as cluster interconnects	Use of ATM as a cluster interconnect improves performance, scalability, and manageability. In addition, a new management utility, the SCA Control Program (SCACP), performs certain privileged cluster management functions.
	Warranted and migration support	OpenVMS Alpha Version 7.3 and OpenVMS VAX Version 7.3 provide two levels of support, warranted and migration, for mixed-version and mixed-architecture OpenVMS Cluster systems.
POLYCENTER Software Installation Utility qualifier	The /FULL qualifier has been added to the PRODUCT LIST command. Use of this qualifier produces output in 132-column format and displays supplemental information about each file in the product kit, such as the size of the file and remarks about its usage.	
Symmetric Multi-Processing (SMP) performance improvements	OpenVMS Alpha Version 7.3 contains software changes that improve SMP scaling.	
SYSMAN commands and qualifiers	Several new SYSMAN commands and qualifiers have been added for OpenVMS Version 7.3.	
System parameters	Several new and updated system parameters have been added for OpenVMS Version 7.3.	

(continued on next page)

Table 1–1 (Cont.) Summary of OpenVMS VAX and OpenVMS Alpha Version 7.3 Software Features

System Management Features	
Volume Shadowing for OpenVMS	Volume Shadowing for OpenVMS introduces three new features: • Minicopy operation, which is a streamlined copy operation. Minicopy can significantly decrease the time it takes to perform a full copy operation and can significantly increase the availability of the shadow sets that use this feature. • New qualifiers added to the DISMOUNT and SET commands for disaster tolerant support for OpenVMS Cluster systems. • A new /SHADOW qualifier to the INITIALIZE command.

Programming Features	
3D graphics support	The PowerStorm 300/350 (PBXGD-AD, PBXGD-AE) graphics cards are now supported on Alpha based systems. The OpenGL 3D graphics API is now provided as part of the base operating system.
ATM LAN adapters (Alpha)	Compaq OpenVMS Version 7.3 provides two new ATM LAN adapters, the 3X-DAPBA-FA and 3X-DAPCA-FA.
Compaq COBOL Run-Time Library	The following enhancements have been made to the COBOL RTL: • Five new intrinsic functions with four-digit year formats • Improved performance for DISPLAY redirected to a file and for programs compiled with /MATH=CIT3 and MATH=CIT4 (Alpha) • Improved compatibility of Alpha and VAX in the handling of the ON SIZE ERROR
Compaq C Run-Time Library enhancements	The following enhancements have been made to the Compaq C Run-Time Library for OpenVMS Version 7.3: • The strptime function was made XPG5-compliant. • The limitation of eight nested directory levels was lifted for an ODS-5 device (Alpha). • Case preservation in file name is supported (Alpha). • Most C RTL functions now accept long OpenVMS-style file names as arguments. • Exact-case argv arguments is supported (Alpha). • Files can be implicitly opened for shared access. • There is an alternative way of translating UNIX file specifications. • Several new C RTL functions have been added.
Fortran 64-bit address support (Alpha)	Support has been added for Fortran developers to use static data in 64-bit address space.

(continued on next page)

Table 1–1 (Cont.) Summary of OpenVMS VAX and OpenVMS Alpha Version 7.3 Software Features

Programming Features	
Large Page-File sections (Alpha)	Previous limits for page-file sections have been extended significantly to take advantage of larger physical memory. Now images that use 64-bit addressing can map and access an amount of dynamic virtual memory that is larger than the amount of physical memory available on the system.
Multipath system services	New services have been added that allow you to return path information and enable, disable, and switch specific I/O paths to any device.
Multiprocess debugging (Alpha)	Debugger support for multiprocess programs has been extensively overhauled. Problems have been corrected and the user-interface has been improved.
Nonpaged executive (Alpha)	In previous releases, some of the OpenVMS Alpha executive code was nonpageable (it resided permanently in memory) and some was pageable. As of OpenVMS Alpha Version 7.3, the base operating system images contain nonpageable code only. This new nonpageable design improves the performance of system services from 5 to 20 percent and uses almost no extra memory because of the method used to link the images.
Performance API	The Performance Application Programming Interface (API) provides a documented functional interface—the $GETRMI system service—that allows performance software engineers to access a predefined list of performance data items.
POLYCENTER Software Installation Utility enhancements	Enhancements have been made to the POLYCENTER Software Installation Utility product description language and the *POLYCENTER Software Installation Utility Developer's Guide*.
Process dump tools (Alpha)	New dump format and tools for analyzing process dumps allow you to do the following consistently and reliably: Transport a dump to another system for analysis Analyze threaded and 64-bit processes Analyze images with shared linkages

(continued on next page)

Table 1–1 (Cont.) Summary of OpenVMS VAX and OpenVMS Alpha Version 7.3 Software Features

Programming Features	
RMS locking enhancements	RMS includes the following enhancements for OpenVMS Version 7.3: • Global buffer read-mode bucket locking (Alpha)—Improves RMS application performance on OpenVMS Alpha systems by minimizing locking for shared access to global buffers. Does not require changes to existing applications. • No query record locking (Alpha)—Improves RMS application performance on OpenVMS Alpha systems by allowing applications to read records without locking them. Avoids the processing associated with record locking calls to the lock manager. Can be enabled on a per-record read operation or at the file level. Includes the new /QUERY_LOCK qualifier and values to the SET RMS_DEFAULT command to disable query record locking at the process level. • Record Locking options to control deadlock detection—Allow RMS to directly control deadlock detection. Options correspond to existing lock request flags used by the distributed lock manager (through $ENQ system service calls).
OpenVMS Registry	The OpenVMS Registry includes the ability to specify the number of seconds that the $REGISTRY service will wait for a response from the Registry Server.
System Dump Analyzer (SDA) commands, parameters, and qualifiers (Alpha)	New Alpha SDA commands, parameters, and qualifiers expand the functionality of the System Dump Analysis tool.
System Dump Analyzer (SDA) commands for the Spinlock Tracing utility (Alpha)	The new SDA Spinlock Tracing utility has commands for gathering and analyzing spinlock performance data. The SDA Spinlock Tracing utility was introduced in OpenVMS Alpha Version 7.2-1H1.
System services	Several system services have been added and updated for Version 7.3.
TCP/IP files for Alpha and VAX SDA	TCP/IP files contain the global symbols and locations within the Executive Image for the Alpha and VAX SDA READ command.
Visual Threads Version 2.1 (Alpha)	The Visual Threads diagnostic tool helps you analyze and refine your multithreaded applications. You can use it to debug potential thread-related problems and to pinpoint bottlenecks and performance problems. Visual Threads Version 2.1 contains functionality and enhancements to help you find application problems before they occur.
Associated Products Features	
Availability Manager Version 1.4	OpenVMS Version 7.3 contains Availability Manager Version 1.4. Soon after the release of OpenVMS Version 7.3, Availability Manager Version 2.0 will become available, which includes several new features and enhancements.
Compaq Advanced Server V7.3 for OpenVMS (Alpha)	Compaq Advanced Server V7.3 contains new functionality and enhancements.
Compaq BASIC for OpenVMS Alpha enhancements	Compaq BASIC now supports the IEEE floating-point data types.

(continued on next page)

Table 1–1 (Cont.) Summary of OpenVMS VAX and OpenVMS Alpha Version 7.3 Software Features

Associated Products Features	
Compaq DECwindows Motif for OpenVMS	Compaq DECwindows Motif for OpenVMS Version 1.2-6 provides a full range of changes and enhancements for your desktop.
Compaq Distributed Computing Environment (DCE) for OpenVMS	The following enhancements are available in Compaq DCE for OpenVMS Version 7.3: • The NT Lan Manager security in DCE RPC is fully functional. • New Ethernet devices may be defined with the system logical DCE$IEEE_802_DEVICE.
DECram Version 3.0 (Alpha)	DECram Version 3.0 contains the following performance enhancements for Galaxy customers: • Capability to use Galaxy shared memory to create an OpenVMS shared memory disk • Faster access for data in a shared disk environment DECram Version 3.0 and supporting documentation are included in the OpenVMS Version 7.3 CD-ROM in the [.DECRAM_030] directory.
Enterprise Capacity and Performance (ECP)	Beginning with OpenVMS Version 7.3, the performance management tools ECP Data Collector for OpenVMS and the ECP Performance Analyzer for OpenVMS are included with the operating system.
Kerberos for OpenVMS	Kerberos is a network authentication protocol designed to provide strong authentication for client/server applications by using secret-key cryptography.
LDAP API (Alpha)	The LDAP API is an addition to OpenVMS that provides the capability to access remote directory services such as X.500, ActiveDirectory or Novell's NDS using the Lightweight Directory Access Protocol (LDAP). Compaq provides the tools to create a client or add LDAP functionality to an existing application so that it can access remote directories.
Compaq PATHWORKS for OpenVMS (Advanced Server)	OpenVMS Version 7.3 supports Compaq PATHWORKS V6.0D for OpenVMS (Advanced Server). Earlier versions of PATHWORKS for OpenVMS servers must be upgraded.
Compaq Service Tools and DECevent	The Compaq Services new web-based service tool functionality is known as Web-Based Enterprise Services (WEBES). The Compaq System Tools CD-ROM included in the OpenVMS Version 7.3 CD-ROM package includes WEBES. (WEBES includes the Compaq Crash Analysis Tool (CCAT) and Compaq Analyze components.) In addition to WEBES, the Compaq System Tools CD-ROM includes DECevent, DSNLINK, and the Revision and Configuration Management (RCM) tools.
Compaq TCP/IP Services for OpenVMS Version 5.1	Compaq TCP/IP Services contains new functionality and enhancements for OpenVMS Version 7.3.

2

Compaq OpenVMS e-Business Features

This chapter provides information on the e-Business technologies that are included in the Compaq OpenVMS e-Business Infrastructure Package with OpenVMS Alpha Version 7.3. This package provides key Internet and e-Business software technology that enhances the base OpenVMS Alpha operating system. These technologies are licensed with the OpenVMS Alpha operating system.

The Compaq OpenVMS e-Business Infrastructure Package Version 1.1 contains the following software and accompanying documentation:

- Compaq Secure Web Server for OpenVMS Alpha Version 1.0-1 (based on Apache)
- Compaq COM for OpenVMS Version 1.1B
- Compaq Java 2 SDK, Standard Edition v 1.2.2-3
- Compaq Fast Virtual Machine (Fast VM) for the Java 2 Platform on OpenVMS Alpha v 1.2.2-1
- Compaq XML (Extensible Markup Language) Technology Version 1.0
- Attunity Connect "On Platform" Package Version 3.0.0.4
- Compaq Enterprise Directory Services for e-Business Version 5.0
- Reliable Transaction Router (RTR) Version 4.0
- Compaq BridgeWorks Version 1.0A

Refer to the *Compaq OpenVMS e-Business Infrastructure Package Version 1.1 CD-ROM Booklet* and the *Compaq OpenVMS e-Business Infrastructure Package Version 1.1 Software Product Description, 80.58.00* included in the e-Business package for more detailed information.

For up-to-date information on OpenVMS e-Business technologies, refer to the following web site:

`http://www.openvms.compaq.com/business/index.html`

The following sections briefly describe the e-Business software and provide pointers and web sites for further information. Refer to the *Compaq OpenVMS e-Business Infrastructure Package SPD* for technology descriptions, other software requirements, and licensing information. The technology on the e-Business CD-ROM has been tested and qualified with OpenVMS Alpha Version 7.2-1 and later.

2.1 Compaq Secure Web Server for OpenVMS Alpha

Compaq Secure Web Server for OpenVMS Alpha (CSWS) is based on the popular Apache Web Server from the Apache Software Foundation. Building on the source code from the Apache Software Foundation (http://www.apache.org), Compaq OpenVMS engineering has incorporated and fully integrated OpenSSL with mod_ssl, the most popular open-source implementations of SSL.

The product is also available to download from the CSWS web site:

`http://www.openvms.compaq.com/openvms/products/ips/apache/csws.html`

2.2 Compaq COM for OpenVMS

Component Object Model (COM) is a technology from Microsoft that allows developers to create distributed network objects. Digital Equipment Corporation and Microsoft jointly developed the COM specification. The Compaq COM for OpenVMS kit included on the e-Business CD-ROM provides all the code and documentation you need to install Compaq COM for OpenVMS on your system and to develop COM applications.

2.3 Compaq Java 2 SDK, Standard Edition, for OpenVMS Alpha

The Java Software Development Kit (SDK) provides an environment in which to develop and deploy Java applications on OpenVMS Alpha. Java applications can be written once and run on any operating system that implements the Java run-time environment, which consists primarily of the Java Virtual Machine (JVM).

The Java 2 SDK, Standard Edition, for OpenVMS Alpha kit is included on the e-Business CD-ROM, or you can download this kit from the Compaq Java home page at the following web address:

`http://www.compaq.com/java/download/index.html`

2.4 Compaq Fast Virtual Machine (Fast VM) for the Java 2 Platform

The Compaq Fast VM for Java 2 is new Just-In-Time (JIT) compiler technology designed to provide optimal Java run-time performance on OpenVMS Alpha systems. The Fast VM for Java 2 offers significant performance advantages over the Classic JIT provided with the Compaq Java 2 SDK, Standard Edition.

The Fast VM for OpenVMS Alpha kit is included on the e-Business CD-ROM, or you can also download this kit from the Compaq Java home page at the following web address:

`http://www.compaq.com/java/download/index.html`

2.5 Compaq XML Technology

The following components are provided on the e-Business CD-ROM using open source software from the Apache Software Foundation:

- XML parsers in Java and C++
- XSLT style sheet processors in Java and C++

This technology provides applications the ability to parse, generate, manipulate, validate, and transform Extensible Markup Language (XML) documents and data.

2.6 Attunity Connect "On Platform" Package

Attunity Connect is object-oriented middleware that facilitates the development of applications that access, integrate, and update data from multiple, heterogeneous sources across a wide range of operating system platforms. With Attunity Connect, you can extend the life of your existing data and applications and preserve your significant IT investments.

The e-Business CD-ROM contains the Attunity Connect "On Platform" Package for OpenVMS Alpha. You can also download the Attunity Connect "On Platform" Package from the following OpenVMS web site:

`http://www.openvms.compaq.com/openvms/products/ips/attunity/`

2.7 Compaq Enterprise Directory Services for e-Business

Compaq OpenVMS Enterprise Directory for e-Business combines the best of both industry standard LDAPv3 and X.500 capabilities, and delivers robust and scalable directory services across intranets, extranets, and the Internet to customers, suppliers and partners. Lightweight Directory Access Protocol (LDAP) support allows access by a myriad of LDAP-based clients, user agents, and applications. The X.500 support brings high performance, resilience, advanced access controls, and easy replication across the enterprise.

For further information, refer to the *Compaq OpenVMS Enterprise Directory for e-Business Software Product Description* (SPD 40.77.*xx*) included on the e-Business CD-ROM in the Enterprise Directory Services documentation directory.

2.8 Reliable Transaction Router (RTR)

Reliable Transaction Router (RTR) is fault tolerant transactional messaging middleware used to implement large, distributed applications using client/server technology. Reliable Transaction Router enables computing enterprises to deploy distributed applications on OpenVMS Alpha and VAX systems.

Refer to the *Reliable Transaction Router for OpenVMS Software Product Description* (SPD 51.04.*xx*) included on the e-Business CD-ROM for additional information; or you can access the RTR web site at:

`http://www.compaq.com/rtr/`

2.9 Compaq BridgeWorks

Compaq BridgeWorks is a distributed application development and deployment tool for OpenVMS 3GL applications. BridgeWorks consists of a GUI development tool on the Windows NT desktop, a server manager component on OpenVMS, and extensive online help. BridgeWorks provides developers with an easy means to create distributed applications using OpenVMS as the enterprise server and Windows NT as the departmental server.

For more information on Compaq BridgeWorks, refer to the *Compaq OpenVMS e-Business Infrastructure Package Software Product Description*.

3
General User Features

This chapter provides new features information for all users of the Compaq OpenVMS Operating System.

3.1 DCL Commands and Lexical Functions

This section describes new and changed DCL commands, qualifiers, and lexical functions for OpenVMS Version 7.3. The following table contains a summary of these changes.

For more information, refer to the *OpenVMS DCL Dictionary*.

DCL Command	Documentation Update
ANALYZE/IMAGE	A new qualifier, /SELECT, has been added, along with an example.
ANALYZE/OBJECT	A new qualifier, /SELECT, has been added, along with an example.
ANALYZE/PROCESS	A new qualifier, /[NO]IMAGE_PATH, has been added, along with an example.
DELETE	A new qualifier, /BITMAP, has been added to support Write Bitmap.
DELETE/INTRUSION	A new qualifier, /NODE, has been added, along with an example, to support Cluster-wide Intrusion.
DIRECTORY	A new qualifier, /CACHING_ATTRIBUTE, has been added to support Extended File Cache (XFC).
DISMOUNT	A new qualifier, /POLICY, has been added to support Write Bitmap. A new qualifier, /FORCE_REMOVAL, has been added to support Volume Shadowing.
DUMP	A new qualifier, /PROCESS, has been added.
INITIALIZE	The INITIALIZE description has been updated to include information about Extended File Cache (XFC). A new qualifier, /SHADOW, has been added to support Volume Shadowing.
MOUNT	The MOUNT command has been moved to the *OpenVMS DCL Dictionary* from the *OpenVMS System Management Utilities Reference Manual*. The MOUNT description has been updated to include information about Extended File Cache (XFC). A new qualifier, /POLICY, has been added to support Write Bitmap.

General User Features
3.1 DCL Commands and Lexical Functions

DCL Command	Documentation Update
SET AUDIT	A new keyword, SERVER, has been added under the LOGFAILURE, LOGIN, and LOGOUT keywords.
	New text has been added to the /NEW_LOG qualifier.
SET CACHE/RESET	This new DCL command has been added to support Extended File Cache (XFC).
SET DEVICE	The following new qualifiers have been added to support Volume Shadowing: /FORCE_REMOVAL, /MEMBER_ TIMEOUT, /MVTIMEOUT, /READ_COST, /SITE, /COPY_ SOURCE, /ABORT_VIRTUAL_UNIT.
SET DISPLAY	The logical, DECW$SETDISPLAY_DEFAULT_ TRANSPORT, has been added to this command.
SET FILE	Two new qualifiers, /SHARE and /CACHING_ATTRIBUTE, have been added. The /CACHING_ATTRIBUTE qualifier supports Extended File Cache (XFC).
SET PROCESS	The functionality of the qualifier, /[NO]DUMP, has been extended to include other processes. The /DUMP qualifier also has a new option, NOW, to initiate an immediate dump of another process.
SET RMS_DEFAULT	Two new qualifiers, /CONTENTION_POLICY and /QUERY_ LOCK have been added, and the examples have been updated.
SET SERVER	Added support for the Registry, including new qualifiers and examples.
SET VOLUME	A new qualifier, /[NO]WRITETHROUGH, has been added to support Extended File Cache (XFC).
	The /HIGHWATER qualifier is valid for Files-11 On-Disk Structure Level 5 disks.
SHOW CPU	The following new qualifiers have been added: /EXACT, /HIGHLIGHT, /OUTPUT, /PAGE, /SEARCH, and /WRAP.
SHOW DEVICES	A new qualifier, /BITMAP, has been added to support Write Bitmap, along with examples.
	The /FULL qualifier now displays the worldwide identifier (WWID) for Fibre Channel tape devices.
SHOW INTRUSION	A new qualifier, /NODE, has been added, along with an example, to support Cluster-wide Intrusion.
SHOW LICENSE	The qualifier, /CHARGE_TABLE, has been added as a synonym for the /UNIT_REQUIREMENTS qualifier.
SHOW MEMORY	The /CACHE qualifier and examples have been updated for Extended File Cache (XFC).
	The /FILES and /FULL qualifiers and examples have been updated for Large Page Files.
SHOW RMS_DEFAULT	The example has been updated.
SHOW SERVER	This command has been added in support of the Registry.
UNLOCK	This command is now obsolete. Use the SET FILE/UNLOCK command.

DCL Lexical	Documentation Update
F$GETDVI	The item codes, MT3_DENSITY, MT3_SUPPORTED, and WWID have been added, and the MOUNTCNT item code has been updated. The item codes, DEVTYPE, DEVCLASS, and DEVICE_ TYPE_NAME have been updated, and an example has been added. Tables 1-7 and 1-8 have been removed.
F$GETQUI	The JOB_STATUS item code list has been updated.
F$GETJPI	The MULTITHREAD item code has been added.
F$GETSYI	The MULTITHREAD and DECNET_VERSION items have been added.

3.2 New Online Help Topics for Utility Routines

As of Version 7.3, online help now includes all the OpenVMS utility routines that are described in *OpenVMS Utility Routines Manual*, including the following:

ACL_Editor
BACKUP_API
CLI_Routines
CONV$_Routines
CQUAL_Routines
DCX_Routines
DECTPU
EDT_Routines
FDL_Routines
LBR_Routines
LDAP_Routines
LGI_Routines
MAIL_Routines
NCS_Routines
PSM_Routines
SMB_Routines
SOR_Routines

For OpenVMS Version 7.3, several online help topics have been renamed, as follows:

Old Topic Name	New Topic Name
BACKUP	BACKUP_Command
FDL	FDL_Files
MAIL	MAIL_Command
NCS	NCS_Command

3.3 MIME Utility Enhancements

The following new commands and qualifiers have been added to the Multipurpose Internet Mail Extension (MIME) utility:

Command	Description
ADD/BINARY	Sets the Content-Type to application/octet-stream and Content-Transfer-Encoding to Base64. This format can be used to represent an arbitrary binary data stream.
SHOW option	Displays information about the MIME environment. Possible options are CONTENT_TYPE, FILE_TYPES, and VERSION.

For more information about the MIME utility commands and qualifiers, refer to the *OpenVMS User's Manual*.

3.4 WWPPS Utility (Alpha)

The World-Wide PostScript Printing Subsystem (WWPPS) is a utility that allows you to print a PostScript file with various language characters on any PostScript printer. By embedding font data within the PostScript printable file, you can print the language characters even if the printer does not have the local language character fonts.

For detailed instructions about using the WWPPS utility, refer to the *OpenVMS User's Manual*.

For more information about the installation and administration of the WWPPS utility, refer to the *OpenVMS System Manager's Manual*.

4
System Management Features

This chapter provides information about new features, changes, and enhancements for system managers.

4.1 OpenVMS Support for New AlphaServer GS Series Systems

OpenVMS Alpha Version 7.3 provides support for Compaq's AlphaServer GS80, GS160 and GS320 systems, which was introduced in OpenVMS Version 7.2-1H1, and includes:

- OpenVMS support for hard and soft partitions (Galaxy) on AlphaServer GS160 and GS320 systems
- OpenVMS Resource Affinity Domain (RAD) support for applications
- CPU Online Replace

4.1.1 OpenVMS Support for Hard and Soft Partitions

Hard partitioning is a physical separation of computing resources by hardware-enforced access barriers. It is impossible to read or write across a hard partition boundary. There is no resource sharing between hard partitions.

Soft partitioning is a separation of computing resources by software-controlled access barriers. Read and write access across a soft partition boundary is controlled by the operating system. OpenVMS Galaxy is an implementation of soft partitioning.

The way customers choose to partition their systems depends on their computing environments and application requirements. For more information about using hard partitions and OpenVMS Galaxy, refer to the *OpenVMS Alpha Partitioning and Galaxy Guide*.

4.1.2 OpenVMS Resource Affinity Domain (RAD) Support for Applications

OpenVMS Alpha Version 7.3 provides non-uniform memory awareness (NUMA) in OpenVMS memory management and process scheduling, which was introduced in OpenVMS Version 7.2-1H1. This capability provides application support for resource affinity domains (RADs), to ensure that applications running on a single instance of OpenVMS on multiple quad building blocks (QBBs) can execute as efficiently as possible in a NUMA environment. A RAD is a set of hardware components (CPU, memory, IO) with common access characteristics, and corresponds to a QBB in an AlphaServer GS160 or GS320 system.

For more information about using the OpenVMS RAD support for application features, refer to the *OpenVMS Alpha Partitioning and Galaxy Guide*.

4.1.3 OpenVMS Support for CPU Online Replace

With OpenVMS Alpha Version 7.3, you can replace secondary CPUs on a running system without rebooting, which provides increased system maintainability and serviceability. This feature is supported only on AlphaServer GS160/320 systems. Note that replacing the primary CPU requires rebooting.

To use this feature, you must first download console firmware Version 5.9B from the following location:

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

After you upgrade the console with the latest firmware, you can then use the following DCL commands to replace a CPU without rebooting:

1. Direct OpenVMS to stop scheduling processes on the CPU:

   ```
   $ STOP/CPU n
   ```

   (*n* is the number of the CPU to be stopped.)

2. Power off the running CPU:

   ```
   $ SET CPU/POWER=OFF n
   ```

3. When the light on the CPU module has turned from green to amber, physically remove the CPU module from the system. Then put in a new CPU.

4. Power on the CPU:

   ```
   $ SET CPU/POWER=ON  n
   ```

OpenVMS automatically adds the CPU to the active set of processors.

Note that the Galaxy Configuration Utility (GCU) also supports this capability.

4.2 Automatic Setting of Daylight Savings Time

System parameter AUTO_DLIGHT_SAV controls whether OpenVMS will automatically change system time to and from Daylight Savings Time when appropriate. A value of 1 tells OpenVMS to automatically make the change. The default is 0 (off). This is a static parameter.

However, if you have a time service (such as DTSS), that time service continues to control time changes, and OpenVMS does not interfere. Do not enable automatic daylight savings time if you have another time service.

For more information, refer to the *OpenVMS System Manager's Manual*.

4.3 Class Scheduler for CPU Scheduling

With OpenVMS Version 7.3, there is a new SYSMAN-based interface for class scheduling. This new class scheduler, implemented on both VAX and Alpha systems, gives you the ability to designate the amount of CPU time that a system's users may receive by placing the users into scheduling classes. Each class is assigned a percentage of the overall system's CPU time. As the system runs, the combined set of users in a class are limited to the percentage of CPU execution time allocated to their class. The users may get some additional CPU time if */windfall* is enabled for their scheduling class. Enabling the */windfall* allows the system to give a small amount of CPU time to a scheduling class when a CPU is idle and the scheduling class' allotted time has been depleted.

To invoke the class scheduler, you use the SYSMAN interface. SYSMAN allows you to create, delete, modify, suspend, resume, and display scheduling classes. Table 4–1 shows the SYSMAN command, CLASS_SCHEDULE, and its sub-commands.

Table 4–1 SYSMAN command: class_schedule

Sub-command	Meaning
ADD	Creates a new scheduling class
DELETE	Deletes a scheduling class
MODIFY	Modifies the characteristics of a scheduling class
SHOW	Shows the characteristics of a scheduling class
SUSPEND	Suspends temporarily a scheduling class
RESUME	Resumes a scheduling class

By implementing the class scheduler using the SYSMAN interface, you create a permanent database that allows OpenVMS to class schedule processes automatically after a system has been booted and rebooted. This database resides on the system disk in SYS$SYSTEM:VMS$CLASS_SCHEDULE.DATA. SYSMAN creates this file as an RMS indexed file when the first scheduling class is created by the SYSMAN command, CLASS_SCHEDULE ADD.

In a cluster environment, SYSMAN creates this database file in the SYS$COMMON root of the [SYSEXE] directory. As a result, the database file is shared among all cluster members. By using SYSMAN's SET ENVIRONMENT command, you can define scheduling classes either on a cluster-wide or per-node basis.

If desired, a system manager (or application manager) uses the permanent class scheduler to place a process into a scheduling class at process creation time. When a new process is created, Loginout determines whether this process belongs to a scheduling class. Given process information from the SYSUAF file, Loginout then class schedules the process if Loginout determines that the process belongs to a scheduling class.

By using the SYSMAN utility to perform class scheduling operations instead of $SCHED system service, you gain the following benefits:

- You need not modify individual program images to control class scheduling. You can add, delete, and modify scheduling classifications from the SYSMAN utility.
- You can use SYSMAN to create a permanent class scheduling database file which allows processes to be class scheduled at process creation time and allows class definitions to be preserved in case of a system reboot.

For more detailed information, refer to the following manuals:

OpenVMS Programming Concepts Manual, Volume I
OpenVMS DCL Dictionary: N–Z
OpenVMS System Services Reference Manual: A–GETUAI

4.4 Dedicated CPU Lock Manager (Alpha)

The Dedicated CPU Lock Manager is a new feature that improves performance on large SMP systems that have heavy lock manager activity. The feature dedicates a CPU to performing lock manager operations.

A dedicated CPU has the following advantages for overall system performance as follows:

- Reduces the amount of MP_SYNCH time
- Provides good CPU cache utilization

4.4.1 Implementing the Dedicated CPU Lock Manager

For the Dedicated CPU Lock Manager to be effective, systems must have a high CPU count and a high amount of MP_SYNCH due to the lock manager. Use the MONITOR utility and the MONITOR MODE command to see the amount of MP_SYNCH. If your system has more than five CPUs and if MP_SYNCH is higher than 200%, your system may be able to take advantage of the Dedicated CPU Lock Manager. You can also use the spinlock trace feature in the System Dump Analyzer (SDA) to help determine if the lock manager is contributing to the high amount of MP_SYNCH time.

The Dedicated CPU Lock Manager is implemented by a LCKMGR_SERVER process. This process runs at priority 63. When the Dedicated CPU Lock Manager is turned on, this process runs in a compute bound loop looking for lock manager work to perform. Because this process polls for work, it is always computable; and with a priority of 63 the process will never give up the CPU, thus consuming a whole CPU.

If the Dedicated CPU Lock Manager is running when a program calls either the $ENQ or $DEQ system services, a lock manager request is placed on a work queue for the Dedicated CPU Lock Manager. While a process waits for a lock request to be processed, the process spins in kernel mode at IPL 2. After the dedicated CPU processes the request, the status for the system service is returned to the process.

The Dedicated CPU Lock Manager is dynamic and can be turned off if there are no perceived benefits. When the Dedicated CPU Lock Manager is turned off, the LCKMGR_SERVER process is in a HIB (hibernate) state. The process may not be deleted once started.

4.4.2 Enabling the Dedicated CPU Lock Manager

To use the Dedicated CPU Lock Manager, set the LCKMGR_MODE system parameter. Note the following about the LCKMGR_MODE system parameter:

- Zero (0) indicates the Dedicated CPU Lock Manager is off (the default).
- A number greater than zero (0) indicates the number of CPUs that should be active before the Dedicated CPU Lock Manager is turned on.

Setting LCKMGR_MODE to a number greater than zero (0) triggers the creation of a detached process called LCKMGR_SERVER. The process is created, and it starts running if the number of active CPUs equals the number set by the LCKMGR_MODE system parameter.

In addition, if the number of active CPUs should ever be reduced below the required threshold by either a STOP/CPU command or by CPU reassignment in a Galaxy configuration, the Dedicated CPU Lock Manager automatically turns off within one second, and the LCKMGR_SERVER process goes into a hibernate state. If the CPU is restarted, the LCKMGR_SERVER process again resumes operations.

4.4.3 Using the Dedicated CPU Lock Manager With Affinity

The LCKMGR_SERVER process uses the affinity mechanism to set the process to the lowest CPU ID other than the primary. You can change this by indicating another CPU ID with the LCKMGR_CPUID system parameter. The Dedicated CPU Lock Manager then attempts to use this CPU. If this CPU is not available, it reverts back to the lowest CPU other than the primary.

The following shows how to dynamically change the CPU used by the LCKMGR_SERVER process:

```
$RUN SYS$SYSTEM:SYSGEN
SYSGEN>USE ACTIVE
SYSGEN>SET LCKMGR_CPUID 2
SYSGEN>WRITE ACTIVE
SYSGEN>EXIT
```

This change applies to the currently running system. A reboot reverts back to the lowest CPU other than the primary. To permanently change the CPU used by the LCKMGR_SERVER process, set LCKMGR_CPUID in your MODPARAMS.DAT file.

To verify the CPU dedicated to the lock manager, use the SHOW SYSTEM command, as follows:

```
$ SHOW SYSTEM/PROCESS=LCKMGR_SERVER
OpenVMS V7.3 on node JYGAL  24-OCT-2000 10:10:11.31  Uptime  3 20:16:56
  Pid    Process Name    State  Pri      I/O       CPU       Page flts  Pages
4CE0021C LCKMGR_SERVER   CUR  2  63        9   3 20:15:47.78        70     84
```

Note that the State field shows the process is currently running on CPU 2.

Compaq highly recommends that a process not be given hard affinity to the CPU used by the Dedicated CPU Lock Manager. With hard affinity when such a process becomes computable, it cannot obtain any CPU time, because the LCKMGR_SERVER process is running at the highest possible real-time priority of 63. However, the LCKMGR_SERVER detects once per second if there are any computable processes that are set by the affinity mechanism to the dedicated lock manager CPU. If so, the LCKMGR_SERVER switches to a different CPU for one second to allow the waiting process to run.

4.4.4 Using the Dedicated CPU Lock Manager with Fast Path Devices

OpenVMS Version 7.3 also introduces Fast Path for SCSI and Fibre Channel Controllers along with the existing support of CIPCA adapters. The Dedicated CPU Lock Manager supports both the LCKMGR_SERVER process and Fast Path devices on the same CPU. However, this may not produce optimal performance.

By default, the LCKMGR_SERVER process runs on the first available nonprimary CPU. Compaq recommends that the CPU used by the LCKMGR_SERVER process not have any Fast Path devices. This can be accomplished in either of the following ways:

- You can eliminate the first available nonprimary CPU as an available Fast Path CPU. To do so, clear the bit associated with the CPU ID from the IO_PREFER_CPUS system parameter.

  For example, let's say your system has eight CPUs with CPU IDs from zero to seven and four SCSI adapters that will use Fast Path. Clearing bit 1 from IO_PREFER_CPUs would result in the four SCSI devices being bound to CPUs 2, 3, 4, and 5. CPU 1, which is the default CPU the lock manager will use, will not have any Fast Path devices.

- You can set the LCKMGR_CPUID system parameter to tell the LCKMGR_SERVER process to use a CPU other than the default. For the above example, setting this system parameter to 7 would result in the LCKMGR_SERVER process running on CPU 7. The Fast Path devices would by default be bound to CPUs 1, 2, 3, and 4.

4.4.5 Using the Dedicated CPU Lock Manager on the AlphaServer GS Series Systems

The new AlphaServer GS Series Systems (GS80, GS160, and the GS320) have NUMA memory characteristics. When using the Dedicated CPU Lock Manager on one of these systems, the best performance is obtained by utilizing a CPU and memory from within a single Quad Building Block (QBB).

For OpenVMS Version 7.3, the Dedicated CPU Lock Manager does not yet have the ability to decide from where QBB memory should be allocated. However, there is a method to preallocate lock manager memory from the low QBB. This can be done with the LOCKIDTBL system parameter. This system parameter indicates the initial size of the Lock ID Table, along with the initial amount of memory to preallocate for lock manager data structures.

To preallocate the proper amount of memory, this system parameter should be set to the highest number of locks plus resources on the system. The command MONITOR LOCK can provide this information. If MONITOR indicates the system has 100,000 locks and 50,000 resources, then setting LOCKIDTBL to the sum of these two values will ensure that enough memory is initially allocated. Adding in some additional overhead may also be beneficial. Setting LOCKIDTBL to 200,000 thus might be appropriate.

If necessary, use the LCKMGR_CPUID system parameter to ensure that the LCKMGR_SERVER runs on a CPU in the low QBB.

4.5 OpenVMS Enterprise Directory for e-Business (Alpha)[1]

OpenVMS Enterprise Directory for e-Business is a massively scalable directory service, providing both X.500 and LDAPv3 services on OpenVMS Alpha with no separate license fee. OpenVMS Enterprise Directory for e-Business provides the following:

- Large percentage of the Fortune 500 already deploy Compaq X.500 Directory Service (the forerunner of OpenVMS Enterprise Directory for e-Business)
- World's first 64-bit directory service
- Seamlessly combines the scalability and distribution features of X.500 with the popularity and interoperability offered by LDAPv3

[1] On OpenVMS VAX a similar service, but without LDAP support and with more limited performance, is still available with Compaq X.500 Directory Service Version 3.1.

- Inherent replication/shadowing features may be exploited to guarantee 100% up-time
- Systems distributed around the world can be managed from a single point
- Ability to store all types of authentication and security certificates across the enterprise accessible from any location
- Highly configurable schema
- In combination with AlphaServer technology and in-memory database delivers market leading performance and low initiation time

For more detailed information, refer to the Compaq OpenVMS e-Business Infrastructure CD-ROM package which is included in the OpenVMS Version 7.3 CD-ROM kit.

4.6 Extended File Cache (Alpha)

The Extended File Cache (XFC) is a new virtual block data cache provided with OpenVMS Alpha Version 7.3 as a replacement for the Virtual I/O Cache.

Similar to the Virtual I/O Cache, the XFC is a clusterwide, file system data cache. Both file system data caches are compatible and coexist in an OpenVMS Cluster.

The XFC improves I/O performance with the following features that are not available with the Virtual I/O Cache:

- Read-ahead caching
- Automatic resizing of the cache
- Larger maximum cache size
- No limit on the number of closed files that can be cached
- Control over the maximum size of I/O that can be cached
- Control over whether cache memory is static or dynamic

For more information, refer to the chapter on Managing Data Caches in the *OpenVMS System Manager's Manual, Volume 2: Tuning, Monitoring, and Complex Systems*.

4.7 /ARB_SUPPORT Qualifier Added to INSTALL Utility (Alpha)

Beginning with OpenVMS Alpha Version 7.3, you can use the /ARB_SUPPORT qualifier with the ADD, CREATE, and REPLACE commands in the INSTALL utility. The ARB_SUPPORT qualifier provides Access Rights Block (ARB) support to products that have not yet been updated the per-thread security Persona Security Block (PSB) data structure.

This new qualifier is included in the INSTALL utility documentation in the *OpenVMS System Management Utilities Reference Manual*.

4.8 MONITOR Utility New Features

The MONITOR utility has two new class names, RLOCK and TIMER, which you can use as follows:

- MONITOR RLOCK: the dynamic lock remastering statistics of a node
- MONITOR TIMER: Timer Queue Entry (TQE) statistics

These enhancements are discussed in more detail in the MONITOR section of the *OpenVMS System Management Utilities Reference Manual* and in the appendix that discusses MONITOR record formats in that manual.

Also in the MONITOR utility, the display screens of MONITOR CLUSTER, PROCESSES/TOPCPU, and SYSTEM now have new and higher scale values. Refer to the *OpenVMS System Management Utilities Reference Manual: M–Z* for more information.

4.9 OpenVMS Cluster Systems

The following OpenVMS Cluster features are discussed in this section:

- Clusterwide intrusion detection
- Fast Path for SCSI and Fibre Channel (Alpha)
- Floppy disks served in an OpenVMS Cluster system (Alpha)
- New Fibre Channel support (Alpha)
- Switched LAN as a cluster interconnect
- Warranted and migration support

4.9.1 Clusterwide Intrusion Detection

OpenVMS Version 7.3 includes clusterwide intrusion detection, which extends protection against attacks of all types throughout the cluster. Intrusion data and information from each system are integrated to protect the cluster as a whole. Member systems running versions of OpenVMS prior to Version 7.3 and member systems that disable this feature are protected individually and do not participate in the clusterwide sharing of intrusion information.

You can modify the SECURITY_POLICY system parameter on the member systems in your cluster to maintain either a local or a clusterwide intrusion database of unauthorized attempts and the state of any intrusion events.

If bit 7 in SECURITY_POLICY is cleared, all cluster members are made aware if a system is under attack or has any intrusion events recorded. Events recorded on one system can cause another system in the cluster to take restrictive action. (For example, the person attempting to log in is monitored more closely and limited to a certain number of login retries within a limited period of time. Once a person exceeds either the retry or time limitation, he or she cannot log in.) The default for bit 7 in SECURITY_POLICY is clear.

For more information on the system services $DELETE_INTRUSION, $SCAN_INTRUSION, and $SHOW_INTRUSION, refer to the *OpenVMS System Services Reference Manual*.

For more information on the DCL commands DELETE/INTRUSION_RECORD and SHOW INTRUSION, refer to the *OpenVMS DCL Dictionary*.

For more information on clusterwide intrusion detection, refer to the *OpenVMS Guide to System Security*.

4.9.2 Fast Path for SCSI and Fibre Channel (Alpha)

Fast Path for SCSI and Fibre Channel (FC) is a new feature with OpenVMS Version 7.3. This feature improves the performance of Symmetric Multi-Processing (SMP) machines that use certain SCSI ports, or FC.

In previous versions of OpenVMS, SCSI and FC I/O completion was processed solely by the primary CPU. When Fast Path is enabled, the I/O completion processing can occur on all the processors in the SMP system. This substantially increases the potential I/O throughput on an SMP system, and helps to prevent the primary CPU from becoming saturated.

See Section 4.12.2 for information about the SYSGEN parameter, FAST_PATH_PORTS, that has been introduced to control Fast Path for SCSI and FC.

4.9.3 Floppy Disks Served in an OpenVMS Cluster System (Alpha)

Until this release, MSCP was limited to serving disks. Beginning with OpenVMS Version 7.3, serving floppy disks in an OpenVMS Cluster system is supported, enabled by MSCP.

For floppy disks to be served in an OpenVMS Cluster system, floppy disk names must conform to the naming conventions for port allocation class names. For more information about device naming with port allocation classes, refer to the *OpenVMS Cluster Systems* manual.

OpenVMS VAX clients can access floppy disks served from OpenVMS Alpha Version 7.3 MSCP servers, but OpenVMS VAX systems cannot serve floppy disks. Client systems can be any version that supports port allocation classes.

4.9.4 New Fibre Channel Support (Alpha)

Support for new Fibre Channel hardware, larger configurations, Fibre Channel Fast Path, and larger I/O operations is included in OpenVMS Version 7.3. The benefits include:

- Support for a broader range of configurations: the lower cost HSG60 controller supports two SCSI buses instead of six SCSI buses supported by the HSG80; multiple DSGGB 16-port Fibre Channel switches enable very large configurations.
- Backup operations to tape, enabled by the new Modular Data Router (MDR), using existing SCSI tape subsystems
- Distances up to 100 kilometers between systems, enabling more configuration choices for multiple-site OpenVMS Cluster systems
- Better performance for certain types of I/O due to Fibre Channel Fast Path and support for larger I/O requests

The following new Fibre Channel hardware has been qualified on OpenVMS Version 7.2-1 and on OpenVMS Version 7.3:

- KGPSA-CA host adapter
- DSGGB-AA switch (8 ports) and DSGGB-AB switch (16 ports)
- HSG60 storage controller (MA6000 storage subsystem)
- Compaq Modular Data Router (MDR)

OpenVMS now supports Fibre Channel **fabrics**. A Fibre Channel fabric is multiple Fibre Channel switches connected together. (A Fibre Channel fabric is also known as cascaded switches.)

Configurations that use Fibre Channel fabrics can be extremely large. Distances up to 100 kilometers are supported in a multisite OpenVMS Cluster system. OpenVMS supports the Fibre Channel SAN configurations described in the *Compaq StorageWorks Heterogeneous Open SAN Design Reference Guide*, available at the following Compaq web site:

`http://www.compaq.com/storage`

Enabling Fast Path for Fibre Channel can substantially increase the I/O throughput on an SMP system. For more information about this new feature, see Section 4.9.2.

Prior to OpenVMS Alpha Version 7.3, I/O requests larger than 127 blocks were segmented by the Fibre Channel driver into multiple I/O requests. Segmented I/O operations generally have lower performance than one large I/O. In OpenVMS Version 7.3, I/O requests up to and including 256 blocks are done without segmenting.

For more information about Fibre Channel usage in OpenVMS Cluster configurations, refer to the *Guidelines for OpenVMS Cluster Configurations*.

4.9.4.1 New Fibre Channel Tape Support (Alpha)

Fibre Channel tape functionality refers to the support of SCSI tapes and SCSI tape libraries in an OpenVMS Cluster system with shared Fibre Channel storage. The SCSI tapes and libraries are connected to the Fibre Channel by a Fibre-to-SCSI bridge known as the Modular Data Router (MDR).

For configuration information, refer to the *Guidelines for OpenVMS Cluster Configurations*.

4.9.5 LANs as Cluster Interconnects

An OpenVMS Cluster system can use several LAN interconnects for node-to-node communication, including Ethernet, Fast Ethernet, Gigabit Ethernet, ATM, and FDDI.

PEDRIVER, the cluster port driver, provides cluster communications over LANs using the NISCA protocol. Originally designed for broadcast media, PEDRIVER has been redesigned to exploit all the advantages offered by switched LANs, including full duplex transmission and more complex network topologies.

Users of LANs for their node-to-node cluster communication will derive the following benefits from the redesigned PEDRIVER:

- Removal of restrictions for using Fast Ethernet, Gigabit Ethernet, and ATM as cluster interconnects
- Improved performance due to better path selection, multipath load distribution, and support of full duplex communication
- Greater scalability
- Ability to monitor, manage, and display information needed to diagnose problems with cluster use of LAN adapters and paths

4.9.5.1 SCA Control Program

The SCA Control Program (SCACP) utility is designed to monitor and manage cluster communications. (SCA is the abbreviation of Systems Communications Architecture, which defines the communications mechanisms that enable nodes in an OpenVMS Cluster system to communicate.)

In OpenVMS Version 7.3, you can use SCACP to manage SCA use of LAN paths. In the future, SCACP might be used to monitor and manage SCA communications over other OpenVMS Cluster interconnects.

This utility is described in more detail in a new chapter in the *OpenVMS System Management Utilities Reference Manual: M–Z.*

4.9.5.2 New Error Message About Packet Loss

Prior to OpenVMS Version 7.3, an SCS virtual circuit closure was the first indication that a LAN path had become unusable. In OpenVMS Version 7.3, whenever the last usable LAN path is losing packets at an excessive rate, PEDRIVER displays the following console message:

```
%PEA0, Excessive packet losses on LAN Path from local-device-name -
 _  to device-name on REMOTE NODE node-name
```

This message is displayed after PEDRIVER performs an excessively high rate of packet retransmissions on the LAN path consisting of the local device, the intervening network, and the device on the remote node. The message indicates that the LAN path has degraded and is approaching, or has reached, the point where reliable communications with the remote node are no longer possible. It is likely that the virtual circuit to the remote node will close if the losses continue. Furthermore, continued operation with high LAN packet losses can result in a significant loss in performance because of the communication delays resulting from the packet loss detection timeouts and packet retransmission.

The corrective steps to take are:

1. Check the local and remote LAN device error counts to see if a problem exists on the devices. Issue the following commands on each node:

   ```
   $ SHOW DEVICE local-device-name
   $ MC SCACP
   SCACP> SHOW LAN device-name
   $ MC LANCP
   LANCP> SHOW DEVICE device-name/COUNT
   ```

2. If device error counts on the local devices are within normal bounds, contact your network administrators to request that they diagnose the LAN path between the devices.

   If necessary, contact your COMPAQ support representative for assistance in diagnosing your LAN path problems.

For additional PEDRIVER troubleshooting information, see Appendix F of the *OpenVMS Cluster Systems* manual.

4.9.6 Warranted and Migration Support

Compaq provides two levels of support, warranted and migration, for mixed-version and mixed-architecture OpenVMS Cluster systems.

Warranted support means that Compaq has fully qualified the two versions coexisting in an OpenVMS Cluster and will answer all problems identified by customers using these configurations.

Migration support is a superset of the Rolling Upgrade support provided in earlier releases of OpenVMS and is available for mixes that are not warranted. Migration support means that Compaq has qualified the versions for use together in configurations that are migrating in a staged fashion to a newer version of OpenVMS VAX or of OpenVMS Alpha. Problem reports submitted against these configurations will be answered by Compaq. However, in exceptional cases, Compaq may request that you move to a warranted configuration as part of answering the problem.

Compaq supports only two versions of OpenVMS running in a cluster at the same time, regardless of architecture. Migration support helps customers move to warranted OpenVMS Cluster version mixes with minimal impact on their cluster environments.

Table 4–2 shows the level of support provided for all possible version pairings.

Table 4–2 OpenVMS Cluster Warranted and Migration Support

	Alpha/VAX V7.3	**Alpha V7.2–*xxx*/ VAX V7.2**	**Alpha/VAX V7.1**
Alpha/VAX V7.3	WARRANTED	Migration	Migration
Alpha V7.2–*xxx*/ VAX V7.2	Migration	WARRANTED	Migration
Alpha/VAX V7.1	Migration	Migration	WARRANTED

In a mixed-version cluster with OpenVMS Version 7.3, you must install remedial kits on earlier versions of OpenVMS. For OpenVMS Version 7.3, two new features, XFC and Volume Shadowing minicopy, cannot be run on any node in a mixed version cluster unless all nodes running earlier versions of OpenVMS have installed the required remedial kit or upgrade. Remedial kits are available now for XFC. An upgrade for systems running OpenVMS Version 7.2-*xx* that supports minicopy will be made available soon after the release of OpenVMS Version 7.3.

For a complete list of required remedial kits, refer to the *OpenVMS Version 7.3 Release Notes*.

4.10 OpenVMS SMP Performance Improvements (Alpha)

OpenVMS Alpha Version 7.3 contains software changes that improve SMP scaling. Designed for applications running on the new AlphaServer GS-series systems, many of these improvements will benefit all customer applications. The OpenVMS SMP performance improvements in Version 7.3 include the following:

- Improved MUTEX Acquisition

  Mutexes are used for synchronization of numerous events on OpenVMS. The most common use of a mutex is for synchronization of the logical names database and I/O base. In releases prior to OpenVMS Alpha Version 7.3, the manipulation of a mutex was completed with the SCHED spinlock held. Because the SCHED spinlock is a heavily used spinlock with high contention on large SMP systems and only a single CPU could manipulate a mutex, bottlenecks often occurred.

  OpenVMS Alpha Version 7.3 changes the way mutexes are manipulated. The mutex itself is now manipulated with atomic instructions. Thus multiple CPUs manipulate different mutexes in parallel. In most cases, the need to acquire the SCHED spinlock has been avoided. In cases where a process

must be placed into a mutex wait state or when mutex waiters must wake up, SCHED will still need to be acquired.

- Improved Process Scheduling

  Changes made to the OpenVMS process scheduler reduce contention on the SCHED spinlock. Prior to OpenVMS Version 7.3, when a process became computable, the scheduler released all IDLE CPUs to attempt to execute the process. On NUMA systems, all idle CPUs in the RAD were released. These idle CPUs competed for the SCHED spinlock, which added to the contention on the SCHED spinlock. As of OpenVMS Version 7.3, the scheduler only releases a single CPU. In addition, the scheduler releases high numbered CPUs first. This has the effect of avoiding scheduling processes on the primary CPU when possible.

  To use the modified scheduler, users must set the system parameter SCH_CTLFLAGS to 1. This parameter is dynamic.

- Improved SYS$RESCHED

  A number of applications and libraries use the SYS$RESCHED system service, which requests a CPU to reschedule another process. In releases prior to OpenVMS Version 7.3, this system service would lock the SCHED spinlock and attempt to reschedule another computable process on the CPU.

  Prior to OpenVMS Version 7.3, when heavy contention existed on the SCHED spinlock, using SYS$RESCHED system increased resources contention. As of OpenVMS Version 7.3, the SYS$RESCHED system service attempts to acquire the SCHED spinlock with a NOSPIN routine. Thus, if the SCHED spinlock is currently locked, this thread will not spin. It will return back to the caller.

- Lock Manager 2000 and 180 improvements

  There are several changes to the lock manager. For OpenVMS Clusters, the lock manager no longer uses IOLOCK8 for synchronization. It now uses the LCKMGR spinlock, which allows locking and I/O operations to occur in parallel.

  Remaster operations can be performed much faster now. The remaster code sends large messages with data from many locks when remastering as opposed to sending a single lock per message.

  The lock manager supports a Dedicated CPU mode. In cases where there is very heavy contention on the LCKMGR spinlock, dedicating a single CPU to performing locking operations provides a much more efficient mechanism.

- Enhanced Spinlock Tracing capability

  The spinlock trace capability, which first shipped in V7.2-1H1, can now trace forklocks. In systems with heavy contention on the IOLOCK8 spinlock, much of the contention occurs in fork threads. Collecting traditional spinlock data only indicates that the fork dispatcher locked IOLOCK8.

  As of OpenVMS Version 7.3, the spinlock trace has a hook in the fork dispatcher code. This allows the trace to report the routine that is called by the fork dispatch, which indicates the specific devices that contribute to heavy IOLOCK8 contention.

- Mailbox driver change

Prior to OpenVMS Version 7.3, the mailbox driver FDT routines called a routine that locked the MAILBOX spinlock and delivered any required attention ASTs. In most cases, this routine did not require any attention ASTs to be delivered. Because the OpenVMS code that makes these calls already has the MAILBOX spinlock locked, the spinlock acquisition was also an unneeded second acquire of the spinlock.

As of OpenVMS Version 7.3, OpenVMS now first checks to see if any ASTs may need to be delivered prior to calling the routine. This avoids both the call overhead and the overhead of relocking the MAILBOX spinlock that was already owned.

4.11 New SYSMAN Commands and Qualifiers

The SYSMAN utility has the following new commands:

- CLASS_SCHEDULE commands

  The class scheduler provides the ability to limit the amount of CPU time that a system's users receive by placing users in scheduling classes.

Command	Description
CLASS_SCHEDULE ADD	Creates a new scheduling class
CLASS_SCHEDULE DELETE	Deletes a scheduling class
CLASS_SCHEDULE MODIFY	Modifies the characteristics of a scheduling class
CLASS_SCHEDULE RESUME	Resumes a scheduling class that has been suspended
CLASS_SCHEDULE SHOW	Displays the characteristics of a scheduling class
CLASS_SCHEDULE SUSPEND	Temporarily suspends a scheduling class

- IO FIND_WWID and IO_REPLACE_WWID (Alpha-only)

  These commands support Fibre Channel tapes, which are discussed in Section 4.9.4.1.

Command	Description
IO FIND_WWID	Detects all previously undiscovered tapes and medium changers
IO REPLACE_WWID	Replaces one worldwide identifier (WWID) with another

- POWER_OFF qualifier for SYSMAN command SHUTDOWN NODE

  The /POWER_OFF qualifier specifies that the system is to power off after shutdown is complete.

For more information, refer to the SYSMAN section of the *OpenVMS System Management Utilities Reference Manual: M–Z.*

4.12 New System Parameters

This section contains definitions of system parameters that are new in OpenVMS Version 7.3.

4.12.1 AUTO_DLIGHT_SAV

AUTO_DLIGHT_SAV is set to either 1 or 0. The default is 0.

If AUTO_DLIGHT_SAV is set to 1, OpenVMS automatically makes the change to and from daylight saving time.

4.12.2 FAST_PATH_PORTS

FAST_PATH_PORTS is a static parameter that deactivates Fast Path for specific drivers.

FAST_PATH_PORTS is a 32-bit mask. If the value of a bit in the mask is 1, Fast Path is disabled for the driver corresponding to that bit. A value of -1 specifies that Fast Path is disabled for all drivers that the FAST_PATH_PORTS parameter controls.

Bit position zero controls Fast Path for PKQDRIVER (for parallel SCSI), and bit position one controls Fast Path for FGEDRIVER (for Fibre Channel). Currently, the default setting for FAST_PATH_PORTS is 0, which means that Fast Path is enabled for both PKQDRIVER and FGEDRIVER.

In addition, note the following:

- CI drivers are not controlled by FAST_PATH_PORTS. Fast Path for CI is enabled and disabled exclusively by the FAST_PATH system parameter.
- FAST_PATH_PORTS is relevant only if the FAST_PATH system parameter is enabled (equal to 1). Setting FAST_PATH to zero has the same effect as setting FAST_PATH_PORTS to -1.

For additional information, see FAST_PATH and IO_PREFER_CPUS.

4.12.3 GLX_SHM_REG

On Galaxy systems, GLX_SHM_REG is the number of shared memory region structures configured into the Galaxy Management Database (GMDB). If you set GLX_SHM_REG to 0, the default number of shared memory regions are configured.

4.12.4 LCKMGR_CPUID (Alpha)

The LCKMGR_CPUID parameter controls the CPU that the Dedicated CPU Lock Manager runs on. This is the CPU that the LCKMGR_SERVER process will utilize if you turn this feature on with the LCKMGR_MODE system parameter.

If the specified CPU ID is either the primary CPU or a nonexistent CPU, the LCKMGR_SERVER process will utilize the lowest nonprimary CPU.

LCKMGR_CPUID is a DYNAMIC parameter.

For more information, refer to the LCKMGR_MODE system parameter.

4.12.5 LCKMGR_MODE (Alpha)

The LCKMGR_MODE parameter controls usage of the Dedicated CPU Lock Manager. Setting LCKMGR_MODE to a number greater than zero (0) indicates the number of CPUs that must be active before the Dedicated CPU Lock Manager is turned on.

The Dedicated CPU Lock Manager performs all locking operations on a single dedicated CPU. This can improve system performance on large SMP systems with high MP_Synch associated with the lock manager.

For more information about usage of the Dedicated CPU Lock Manager, see the *OpenVMS Performance Management* manual.

Specify one of the following:

Value	Description
0	Indicates the Dedicated CPU Lock Manager is off. (The default.)
>0	Indicates the number of CPUs that must be active before the Dedicated CPU Lock Manager is turned on.

LCKMGR_MODE is a DYNAMIC parameter.

4.12.6 NPAGECALC

NPAGECALC controls whether the system automatically calculates the initial size for nonpaged dynamic memory.

Compaq sets the default value of NPAGECALC to 1 only during the initial boot after an installation or upgrade. When the value of NPAGECALC is 1, the system calculates an initial value for the NPAGEVIR and NPAGEDYN system parameters. This calculated value is based on the amount of physical memory in the system.

NPAGECALC's calculations do not reduce the values of NPAGEVIR and NPAGEDYN from the values you see or set at the SYSBOOT prompt. However, NPAGECALC's calculation might increase these values.

AUTOGEN sets NPAGECALC to 0. NPAGECALC should always remain 0 after AUTOGEN has determined more refined values for the NPAGEDYN and NPAGEVIR system parameters.

4.12.7 NPAGERAD (Alpha)

NPAGERAD specifies the total number of bytes of nonpaged pool that will be allocated for Resource Affinity Domains (RADs) other than the base RAD. For platforms that have no RADs, NPAGERAD is ignored. Notice that NPAGEDYN specifies the total amount of nonpaged pool for all RADs.

Also notice that the OpenVMS system might round the specified values higher to an even number of pages for each RAD, which prevents the base RAD from having too little nonpaged pool. For example, if the hardware is an AlphaServer GS160 with 4 RADs:

```
NPAGEDYN = 6291456 bytes
NPAGERAD = 2097152 bytes
```

In this case, the OpenVMS system allocates a total of approximately 6,291,456 bytes of nonpaged pool. Of this amount, the system divides 2,097,152 bytes among the RADs that are not the base RAD. The system then assigns the remaining 4,194,304 bytes to the base RAD.[1]

4.12.8 RAD_SUPPORT (Alpha)

RAD_SUPPORT enables RAD-aware code to be executed on systems that support Resource Affinity Domains (RADs); for example, AlphaServer GS160 systems.

A RAD is a set of hardware components (CPUs, memory, and I/O) with common access characteristics. For more information about using OpenVMS RAD features, refer to the *OpenVMS Alpha Partitioning and Galaxy Guide*.

4.12.9 SHADOW_MAX_UNIT

SHADOW_MAX_UNIT specifies the maximum number of shadow sets that can exist on a node. The setting **must be equal to or greater** than the number of shadow sets you plan to have on a system. Dismounted shadow sets, unused shadow sets, and shadow sets with no write bitmaps allocated to them are included in the total.

This system parameter is not dynamic; that is, a reboot is required when you change the setting.

The default setting on OpenVMS Alpha systems is 500; on OpenVMS VAX systems, the default is 100. The minimum value is 10, and the maximum value is 10,000.

Note that this parameter does not affect the naming of shadow sets. For example, with the default value of 100, a device name such as DSA999 is still valid.

4.12.10 VCC_MAX_IO_SIZE (Alpha)

The dynamic system parameter VCC_MAX_IO_SIZE controls the maximum size of I/O that can be cached by the Extended File Cache. It specifies the size in blocks. By default, the size is 127 blocks.

Changing the value of VCC_MAX_IO_SIZE affects reads and writes to volumes currently mounted on the local node, as well as reads and writes to volumes mounted in the future.

If VCC_MAX_IO_SIZE is 0, the Extended File Cache on the local node cannot cache any reads or writes. However, the system is not prevented from reserving memory for the Extended File Cache during startup if a VCC$MIN_CACHE_SIZE entry is in the reserved memory registry.

VCC_MAX_IO_SIZE is a DYNAMIC parameter.

4.12.11 VCC_READAHEAD (Alpha)

The dynamic system parameter VCC_READAHEAD controls whether the Extended File Cache can use read-ahead caching. Read-ahead caching is a technique that improves the performance of applications that read data sequentially.

[1] The system actually rounds up to an even number of pages on each RAD. In addition, the base RAD is never assigned a value less than the smaller of the value of NPAGEDYN and 4 megabytes.

By default VCC_READAHEAD is 1, which means that the Extended File Cache can use read-ahead caching. The Extended File Cache detects when a file is being read sequentially in equal-sized I/Os, and fetches data ahead of the current read, so that the next read instruction can be satisfied from cache.

To stop the Extended File Cache from using read-ahead caching, set VCC_READAHEAD to 0.

Changing the value of VCC_READAHEAD affects volumes currently mounted on the local node, as well as volumes mounted in the future.

Readahead I/Os are totally asynchronous from user I/Os and only take place if sufficient system resources are available.

VCC_READAHEAD is a DYNAMIC parameter.

4.12.12 WBM_MSG_INT

WBM_MSG_INT is one of three system parameters that are available for managing the update traffic between a master write bitmap and its corresponding local write bitmaps in an OpenVMS Cluster system. (Write bitmaps are used by the volume shadowing software for minicopy operations.) The others are WBM_MSG_UPPER and WBM_MSG_LOWER. These parameters set the interval at which the frequency of sending messages is tested and also set an upper and lower threshold that determine whether the messages are grouped into one SCS message or are sent one by one.

In single-message mode, WBM_MSG_INT is the time interval in milliseconds between assessments of the most suitable write bitmap message mode. In single-message mode, the writes issued by each remote node are, by default, sent one by one in individual SCS messages to the node with the master write bitmap. If the writes sent by a remote node reach an upper threshold of messages during a specified interval, single-message mode switches to buffered-message mode.

In buffered-message mode, WBM_MSG_INT is the maximum time a message waits before it is sent. In buffered-message mode, the messages are collected for a specified interval and then sent in one SCS message. During periods of increased message traffic, grouping multiple messages to send in one SCS message to the master write bitmap is generally more efficient than sending each message separately.

The minimum value of WBM_MSG_INT is 10 milliseconds. The maximum value is -1, which corresponds to the maximum positive value that a longword can represent. The default is 10 milliseconds.

WBM_MSG_INT is a DYNAMIC parameter.

4.12.13 WBM_MSG_LOWER

WBM_MSG_LOWER is one of three system parameters that are available for managing the update traffic between a master write bitmap and its corresponding local write bitmaps in an OpenVMS Cluster system. (Write bitmaps are used by the volume shadowing software for minicopy operations.) The others are WBM_MSG_INT and WBM_MSG_UPPER. These parameters set the interval at which the frequency of sending messages is tested and also set an upper and lower threshold that determine whether the messages are grouped into one SCS message or are sent one by one.

WBM_MSG_LOWER is the lower threshold for the number of messages sent during the test interval that initiates single-message mode. In single-message mode, the writes issued by each remote node are, by default, sent one by one in individual SCS messages to the node with the master write bitmap. If the writes sent by a remote node reach an upper threshold of messages during a specified interval, single-message mode switches to buffered-message mode.

The minimum value of WBM_MSG_LOWER is 0 messages per interval. The maximum value is -1, which corresponds to the maximum positive value that a longword can represent. The default is 10.

WBM_MSG_LOWER is a DYNAMIC parameter.

4.12.14 WBM_MSG_UPPER

WBM_MSG_UPPER is one of three system parameters that are available for managing the update traffic between a master write bitmap and its corresponding local write bitmaps in an OpenVMS Cluster system. (Write bitmaps are used by the volume shadowing software for minicopy operations.) The others are WBM_MSG_INT and WBM_MSG_LOWER. These parameters set the interval at which the frequency of sending messages is tested and also set an upper and lower threshold that determine whether the messages are grouped into one SCS message or are sent one by one.

WBM_MSG_UPPER is the upper threshold for the number of messages sent during the test interval that initiates buffered-message mode. In buffered-message mode, the messages are collected for a specified interval and then sent in one SCS message.

The minimum value of WBM_MSG_UPPER is 0 messages per interval. The maximum value is -1, which corresponds to the maximum positive value that a longword can represent. The default is 100.

WBM_MSG_UPPER is a DYNAMIC parameter.

4.12.15 WBM_OPCOM_LVL

WBM_OPCOM_LVL controls whether write bitmap system messages are sent to the operator console. (Write bitmaps are used by the volume shadowing software for minicopy operations.) Possible values are shown in the following table:

Value	Description
0	Messages are turned off.
1	The default; messages are provided when write bitmaps are started, deleted, and renamed, and when the SCS message mode (buffered or single) changes.
2	All messages for a setting of 1 are provided plus many more.

WBM_OPCOM_LVL is a DYNAMIC parameter.

4.13 Volume Shadowing for OpenVMS

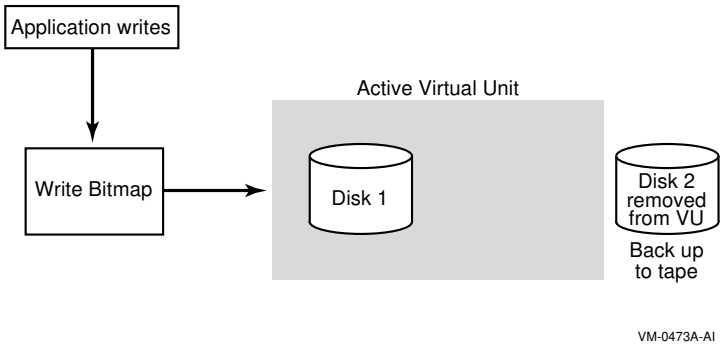
Volume Shadowing for OpenVMS introduces three new features, the minicopy operation enabled by write bitmaps, new qualifiers for disaster tolerant support for OpenVMS Cluster systems, and a new /SHADOW qualifier to the INITIALIZE command. These features are described in this section.

4.13.1 Minicopy in Compaq Volume Shadowing for OpenVMS (Alpha)

The new minicopy feature of Compaq Volume Shadowing for OpenVMS and its enabling technology, write bitmap, are fully implemented on OpenVMS Alpha systems. OpenVMS VAX nodes can write to shadow sets that use this feature but they can neither create master write bitmaps nor manage them with DCL commands.

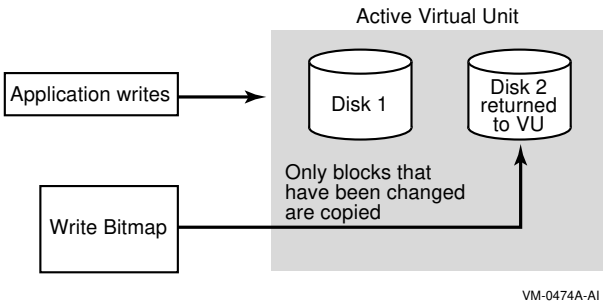
The minicopy operation is a streamlined copy operation. Minicopy is designed to be used in place of a copy operation when you return a shadow set member to the shadow set. When a member has been removed from a shadow set, a write bitmap tracks the changes that are made to the shadow set in its absence, as shown in Figure 4–1.

Figure 4–1 Application Writes to a Write Bitmap



When the member is returned to the shadow set, the write bitmap is used to direct the minicopy operation, as shown in Figure 4–2. While the minicopy operation is taking place, the application continues to read and write to the shadow set.

Figure 4–2 Member Returned to the Shadow Set (Virtual Unit)



Thus, minicopy can significantly decrease the time it takes to return the member to membership in the shadow set and can significantly increase the availability of the shadow sets that use this feature.

Typically, a shadow set member is removed from a shadow set to back up the data on the disk. Before the introduction of the minicopy feature, Compaq required that the virtual unit (the shadow set) be dismounted to back up the data from one of the members. This requirement has been removed, provided that the guidelines for removing a shadow set member for backup purposes, as documented in *Volume Shadowing for OpenVMS*, are followed.

For more information about this new feature, including additional memory requirements for this version of Compaq Volume Shadowing for OpenVMS, refer to *Volume Shadowing for OpenVMS*.

4.13.2 New Volume Shadowing Features for Multiple-Site OpenVMS Cluster Systems

OpenVMS Version 7.3 introduces new command qualifiers for the DCL commands DISMOUNT and SET for use with Volume Shadowing for OpenVMS. These new command qualifiers provide disaster tolerant support for multiple-site OpenVMS Cluster systems. Designed primarily for multiple-site clusters that use Fibre Channel for a site-to-site storage interconnect, they can be used in other configurations as well. For more information about using these new qualifiers in a multiple-site OpenVMS Cluster system, see the white paper Using Fibre Channel in a Disaster-Tolerant OpenVMS Cluster System, which is posted on the OpenVMS Fibre Channel web site at:

`http://www.openvms.compaq.com/openvms/fibre/`

The new command qualifiers are described in this section. Section 4.13.2.1 describes how to use these new qualifiers.

DISMOUNT/FORCE_REMOVAL *ddcu:*

One new qualifier to the DISMOUNT command, DISMOUNT/FORCE_REMOVAL *ddcu:*, is provided. If connectivity to a device has been lost and the shadow set is in mount verification, `/FORCE_REMOVAL` *ddcu:* can be used to immediately expell a named shadow set member (*ddcu:*) from the shadow set. If you omit this qualifier, the device is not dismounted until mount verification completes. Note that this qualifier cannot be used in conjunction with the /POLICY=MINICOPY (=OPTIONAL) qualifier.

The device specified must be a member of a shadow set that is mounted on the node where the command is issued.

SET DEVICE

The following new qualifiers to the SET DEVICE command have been created for managing shadow set members located at multiple sites:

- `/FORCE_REMOVAL` *ddcu:*

  If connectivity to a device has been lost and the shadow set is in mount verification, this qualifier causes the member to be expelled from the shadow set immediately.

  If the shadow set is not currently in mount verification, no immediate action is taken. If connectivity to a device has been lost but the shadow set is not in mount verification, this qualifier lets you flag the member to be expelled from the shadow set, as soon as it does enter mount verification.

  The device specified must be a member of a shadow set that is mounted on the node where the command is issued.

- `/MEMBER_TIMEOUT=`*xxxxxx ddcu:*

  Specifies the timeout value to be used for a member of a shadow set.

  The value supplied by this qualifier overrides the SYSGEN parameter SHADOW_MBR_TMO for this specific device. Each member of a shadow set can be assigned a different MEMBER_TIMEOUT value.

  The valid range for *xxxxxx* is 1 to 16,777,215 seconds.

The device specified must be a member of a shadow set that is mounted on the node where the command is issued.

- /MVTIMEOUT=*yyyyyy* DSA*nnnn*:

  Specifies the mount verification timeout value to be used for this shadow set, specified by its virtual unit name, DSA*nnnn*.

  The value supplied by this qualifier overrides the SYSGEN parameter MVTIMEOUT for this specific shadow set.

  The valid range for *yyyyyy* is 1 to 16,777,215 seconds.

  The device specified must be a shadow set that is mounted on the node where the command is issued.

- /READ_COST=*zzz ddcu:*

  The valid range for *zzz* is 1 to 4,294,967,295 units.

  The device specified must be a member of a shadow set that is mounted on the node where the command is issued.

  This qualifier allows you to modify the default ''cost'' assigned to each member of a shadow set, so that reads are biased or prioritized toward one member versus another.

  The shadowing driver assigns default READ_COST values to shadow set members when each member is initially mounted. The default value depends on the device type, and its configuration relative to the system mounting it. There are default values for a DECRAM device; a directly connected device in the same physical location; a directly connected device in a remote location; a DECram served device; and a default value for other served devices.

  The value supplied by this qualifier overrides the default assignment. The shadowing driver adds the value of the current queue depth of the shadow set member to the READ_COST value and then reads from the member with the lowest value.

  Different systems in the cluster can assign different costs to each shadow set member.

  If the /SITE command qualifier has been specified, the shadowing driver will take site values into account when it assigns default READ_COST values. Note that in order for the shadowing software to determine if a device is in the category of ''directly connected device in a remote location,'' the /SITE command qualifier must have been applied to both the shadow set and to the individual device.

  Reads requested for a shadow set from a system at Site 1 are performed from a shadow set member that is also at Site 1. Reads requested for the same shadow set from Site 2 can read from the member located at Site 2.

- /READ_COST=*y* DSA*nnnn*

  The valid range for *y* is any non-zero number. The value supplied has no meaning in itself. The purpose of this qualifier is to switch the read cost setting for all shadow set members back to the default read cost settings established automatically by the shadowing software. DSA*nnnn* must be a shadow set that is mounted on the node from which this command is issued.

- /SITE=(*nnn, logical_name*) (*ddcu*: DSA*nnnn:*)

This qualifier indicates to the shadowing driver the site location of the shadow set member or of the shadow set (represented by its virtual unit name). Prior to using this qualifier, you can define the site location in the SYLOGICALS.COM command procedure to simplify its use.

The valid range for *nnn* is 1 through 255.

The following example shows the site locations defined, followed by the use of the /SITE qualifier:

```
$ DEFINE/SYSTEM/EXEC ZKO 1
$ DEFINE/SYSTEM/EXEC LKG 2
$!
$! At the ZKO site ...
$ MOUNT/SYSTEM DSA0/SHAD=($1$DGA0:,$1$DGA1:) TEST
$ SET DEVICE/SITE=ZKO  DSA0:
$!
$! At the LKG site ...
$ MOUNT/SYSTEM DSA0/SHAD=($1$DGA0,$1$DGA1) TEST
$ SET DEVICE/SITE=LKG  DSA0:
$!
$! At both sites, the following would be used:
$ SET DEVICE/SITE=ZKO  $1$DGA0:
$ SET DEVICE/SITE=LKG  $1$DGA1:
```

- /COPY_SOURCE (*ddcu*:,DSA*nnnn:*)

  Controls whether one or both source members of a shadow set are used as the source for read data during full copy operations, when a third member is added to the shadow set. This only affects copy operations that do not use DCD operations.

  HSG80 controllers have a read-ahead cache, which significantly improves single-disk read performance. Copy operations normally alternate reads between the two source members, which effectively nullifies the benefits of the read-ahead cache. This qualifier lets you force all reads from a single source member for a copy operation.

  If the shadow set is specified, then all reads for full copy operations will be performed from whichever disk is the current ‘‘master’’ member, regardless of physical location of the disk.

  If a member of the shadow set is specified, then that member will be used as the source of all copy operations. This allows you to choose a local source member, rather than a remote master member.

- /ABORT_VIRTUAL_UNIT DSA*nnnn:*

  To use this qualifier, the shadow set must be in mount verification. When you specify this qualifier, the shadow set aborts mount verification immediately on the node from which the qualifier is issued. This qualifier is intended to be used when it is known that the unit cannot be recovered. Note that after this command completes, the shadow set must still be dismounted. Use the following command to dismount the shadow set:

  ```
  DISMOUNT/ABORT   DSAnnnn
  ```

4.13.2.1 How to Use the New DISMOUNT and SET Command Qualifiers

The diagram in this section depicts a typical multiple-site cluster using Fibre Channel. It is used to illustrate the steps which must be taken to manually recover one site when the site-to-site storage interconnect fails. Note that with current Fibre Channel support, neither site can use the MSCP server to regain a path to the DGA devices.

To prevent the shadowing driver from automatically recovering shadow sets from connection-related failures, three steps must be taken prior to any failure:

1. Every device that is a member of a multiple-site shadow set must have its member_timeout setting raised to a high value, using the following command:

   ```
   $ SET DEVICE /MEMBER_TIMEOUT= x  ddcu:
   ```

   This command will override the SHADOW_MBR_TMO value, which would normally be used for a shadow set member. A value for x of 259200 would be a seventy-two hour wait time.

2. Every shadow set that spans multiple sites must have its mount verification timeout setting raised to a very high value, higher than the MEMBER_ TIMEOUT settings for each member of the shadow set.

   Use the following command to increase the mount verification timeout setting for the shadow set:

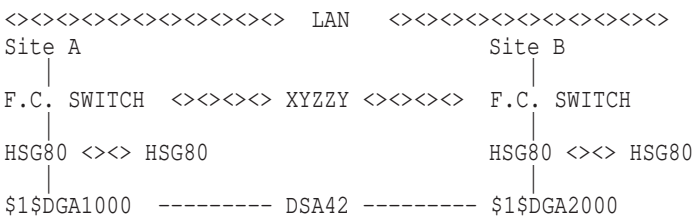
   ```
   $ SET DEVICE /MVTIMEOUT = y  DSAnnnn
   ```

   The y value of this command should always be greater than the x value of the `$ SET DEVICE/MEMBER_TIMEOUT= x ddcu:`.

   The `$ SET DEVICE /MVTIMEOUT = y` command will override the MVTIMEOUT value, which would normally be used for the shadow set. A value for y of 262800 would be a seventy-three hour wait.

3. Every shadow set and every shadow set member must have a site qualifier. As already noted, a site qualifier will ensure that the read cost is correctly set. The other critical factor is three-member shadow sets. When they are being used, the site qualifier will ensure that the master member of the shadow set will be properly maintained.

In the following diagram, shadow set DSA42 is made up of 1DGA1000 and 1DGA2000



This diagram illustrates that systems at Site A or Site B have direct access to all devices at both sites via Fibre Channel connections. XYZZY is a theoretical point between the two sites. If the Fibre Channel connection were to break at this point, each site could access different ‘‘local’’ members of DSA42 without error. For the purpose of this example, Site A will be the sole site chosen to retain access to the shadow set.

The following actions must be taken to recover the shadow set at Site A.

On Site A:

```
$ DISMOUNT /FORCE_REMOVAL= $1$DGA2000:
```

Once the command has completed, the shadow set will be available for use only at site A.

On Site B:

```
$ SET DEVICE /ABORT_VIRTUAL_UNIT DSA42:
```

Once the command completes, the shadow set status will be `MntVerifyTimeout`.

Next, issue the following command to free up the shadow set:

```
$ DISMOUNT/ABORT DSA42:
```

These steps must be taken for all affected multiple-site shadow sets.

4.13.3 Using INITIALIZE/SHADOW/ERASE to Streamline the Formation of a Shadow Set

The new /SHADOW qualifier to the DCL INITIALIZE command is available. The use of the INITIALIZE /SHADOW command to initialize multiple members of a future shadow set eliminates the requirement for a full copy operation when you later create a shadow set.

Compaq *strongly recommends* that you also specify the /ERASE qualifier with the INITIALIZE/SHADOW command when initializing multiple members of a future shadow set. Whereas the /SHADOW qualifier eliminates the need for a full copy operation when you later create a shadow set, the /ERASE qualifier reduces the amount of time a full merge will take.

If you omit the /ERASE qualifier, and a merge operation of the shadow set is subsequently required (because a system on which the shadow set is mounted fails), the resulting merge operation will take much longer to complete.

The INITIALIZE command with the /SHADOW and /ERASE qualifiers performs the following operations:

- Formats up to six devices with one command, so that any three can be subsequently mounted together as members of a new host-based shadow set.
- Writes a label on each volume.
- Deletes all information from the devices except for the system files containing identical file structure information. All former contents of the disks are lost.

You can then mount up to three of the devices that you have initialized in this way as members of a new host-based shadow set.

For more information, refer to *Volume Shadowing for OpenVMS*.

5
Programming Features

This chapter describes new features relating to application and system programming on this version of the Compaq OpenVMS operating system.

5.1 3D Graphics Support

The PowerStorm 300 (PBXGD-AD) and PowerStorm 350 (PBXGD-AE) graphics cards are now supported on Alpha-based systems. The OpenGL 3D graphics API is now provided as part of the OpenVMS base operating system. The version of OpenGL supported on the PowerStorm 300 and PowerStorm 350 graphics cards is Version 1.1.

The implementation of OpenGL Version 1.1 for the PowerStorm 300 or PowerStorm 350 is designed to coexist with installations of the Open3D layered product for older graphics cards. The images shipped with OpenVMS are named DECW$OPENGLSHR_V11 and DECW$OPENGLUSHR_V11. The _V11 suffix is used to distinguish the OpenGL Version 1.1 images from the OpenGL Version 1.0 images shipped with Open3D (DECW$OPENGLSHR and DECW$OPENGLUSHR).

Applications using only OpenGL V1.0 features may be linked against either the Open3D images or the new Version 1.1 images. Applications using OpenGL Version 1.1 features should be linked explicitly against the Version 1.1 images.

For further information on OpenGL support for the PowerStorm 300 and PowerStorm 350, refer to the PowerStorm 300/350 Installation Guide and Release Notes documentation shipped with the graphics card.

WARNING

If 3D graphics will be used extensively, particularly in an environment using multiple PowerStorm 300 and PowerStorm P350s in a single system, read and strictly observe the guidelines for setting SYSGEN parameters and account quotas contained in the *PowerStorm 300/350 OpenVMS Graphics Support Release Notes Version 1.1* and the *Compaq PowerStorm 300/350 Graphics Controllers Installation Guide* shipped with the graphics card. The Release notes can also be accessed on the OpenVMS Documentation CD–ROM in the following directory:

Directory	File Name
[73.DOCUMENTATION.PS_TXT]	P300_350_REL_NOTES.PS,TXT

5.2 3X-DAPBA-FA and 3X-DAPCA-FA ATM LAN Adapters (Alpha)

The 3X-DAPBA-FA (HE155) and 3X-DAPCA-FA (HE622) are PCI based ATM LAN adapters for Alpha based systems that provide high performance PCI-to-ATM capability. The 3X-DAPBA-FA adapter offers a 155 Mbps fiber connection; the 3X-DAPCA-FA adapter offers a 622 Mbps fiber connection.

The datalink drivers for these adapters function in a new OpenVMS ATM environment. The new OpenVMS ATM environment is fully compatible with the existing legacy ATM support and allows both ATM environments to be configured on a single system. Also, the LANCP management interface is the same for both ATM environments.

For additional information about the 3X-DAPBA-FA PCI HE155 ATM and 3X-DAPCA-FA PCI HE622 ATM LAN adapters, refer to the following URL:

`http://www.compaq.com/alphaserver/products/options`

5.3 Compaq COBOL Run-Time Library Enhancements

The COBOL RTL for both Alpha and VAX supports five new intrinsic functions with four-digit year formats:

YEAR-TO-YYYY
DATE-TO-YYYYMMDD
DAY-TO-YYYYDDD
TEST-DATE-YYYYMMDD
TEST-DAY-YYYYDDD

The COBOL RTL for Alpha has improved performance for the DISPLAY statement redirected to a file and for programs compiled with the /MATH=CIT3 and /MATH=CIT4 qualifiers.

This RTL's handling of ON SIZE ERROR is now more compatible with that of Compaq COBOL for OpenVMS VAX.

5.4 Compaq C Run-Time Library Enhancements

The following sections describe the Compaq C RTL enhancements included in OpenVMS Version 7.3. For more details, refer to the revision of the *Compaq C RTL Reference Manual* that ships with Compaq C Version 6.3 or later.

5.4.1 Strptime Function Is XPG5-Compliant

The `strptime` function has been modified to be compliant with X/Open CAE Specification System Interfaces and Headers Issue 5 (commonly known as XPG5). The change for XPG5 is in how the `strptime` function processes the "%y" directive for a two-digit year within the century if no century is specified.

When a century is not otherwise specified, XPG5 requires that values for the "%y" directive in the range 69-99 refer to years in the twentieth century (1969 to 1999 inclusive), while values in the range 00-68 refer to years in the twenty-first century (2000 to 2068 inclusive). Essentially, for the "%y" directive, `strptime` became a "pivoting" function, with 69 being a pivoting year.

Before this change, the `strptime` function interpreted a two-digit year with no century as a year within twentieth century.

With OpenVMS Version 7.3, XPG5-compliant `strptime` becomes a default `strptime` function in the Compaq C RTL. However, the previous nonpivoting XPG4-compliant `strptime` function is retained for compatibility.

The pivoting is controlled by the DECC$XPG4_STRPTIME logical name. To use the nonpivoting version of `strptime`, either:

- Define DECC$XPG4_STRPTIME to any value before invoking the application.

  OR

- Call the nonpivoting `strptime` directly as the function `decc$strptime_xpg4`.

5.4.2 Limitation of Eight Nested Directory Levels Was Lifted (Alpha)

The Compaq C RTL I/O subsystem was enhanced to remove the restriction of eight nested directory levels for an ODS-5 device. This affects Compaq C RTL functions such as `access`, `mkdir`, `opendir`, `rmdir`, and `stat`.

5.4.3 Improved Support for Extended File Specifications (Alpha)

The following sections describe improved Compaq C RTL support for extended file specifications.

5.4.3.1 Compaq C RTL Supports Case Preservation in File Names

Programs linked against the Compaq C Run-Time Library DECC$SHR can now preserve the case of file names on ODS level 5 disks. This applies when creating or reporting file names. By default, this feature is disabled. To enable this feature, enter the following command:

```
$ DEFINE DECC$EFS_CASE_PRESERVE ENABLE
```

If file names are all in uppercase, use the following command to convert the names to lowercase when reporting the name in UNIX style:

```
$ DEFINE DECC$EFS_CASE_SPECIAL ENABLE
```

If file names are not all in uppercase, then DEFINE DECC$EFS_CASE_SPECIAL ENABLE preserves case.

The commands to disable the preceding logical-name settings are:

```
$ DEFINE DECC$EFS_CASE_PRESERVE DISABLE
$ DEFINE DECC$EFS_CASE_SPECIAL DISABLE
```

The setting for the DECC$EFS_CASE_SPECIAL logical name, if not set to DISABLE, supersedes any setting for the DECC$EFS_CASE_PRESERVE logical name.

The DECC$EFS_CASE_PRESERVE and DECC$EFS_CASE_SPECIAL logicals are checked only once per image activation, not on a file-by-file basis.

5.4.3.2 Most C RTL Functions Now Accept Long OpenVMS File Names As Arguments (Alpha)

For OpenVMS Alpha Version 7.2, some basic Compaq C RTL I/O functions (`creat`, `stat`, and the functions from the open family of functions) were enhanced to accept long OpenVMS-style file names for an ODS-5 device.

For OpenVMS Alpha Version 7.3, all other Compaq C RTL functions, except `chdir` and the functions from the `exec` family of functions, were also enhanced to accept long OpenVMS-style file names for an ODS-5 device.

All C RTL functions that accept or report full file specifications will process file specifications up to 4095 bytes long, subject to the rules defined for the media format. For file specifications in OpenVMS format, there are no special restrictions. In situations where a full file specification cannot be reported because the buffer is too short, the function attempts to report the abbreviated name.

UNIX file names have the following restrictions:

- Names containing special characters, such as multiple periods, caret, or multinational characters, may be rejected.
- A function call may report failure if the output buffer is not large enough to receive the full name. For OpenVMS style names, the reported name would contain a file ID-abbreviated name. There is no representation of file ID-abbreviated names defined for UNIX.

5.4.4 Compaq C RTL Supports Exact Case Argv Arguments (Alpha)

Nonquoted command-line arguments passed to C and C++ programs (argv arguments) can now optionally have their case preserved, rather than being lowercased as in previous versions.

By default, this feature is disabled.

To enable this case preservation feature, define the logical name DECC$ARGV_PARSE_STYLE to "ENABLE" and set the process-level DCL parse style flag to "EXTENDED" in the process running the program:

```
$ DEFINE DECC$ARGV_PARSE_STYLE ENABLE
$ SET PROCESS/PARSE_STYLE=EXTENDED
```

Enabling this feature also ensures that the image name returned in argv[0] is also case-preserved.

To disable this feature, use any one of the following commands:

```
 $ SET PROCESS/PARSE_STYLE=TRADITIONAL
```

or

```
$ DEFINE/SYSTEM DECC$ARGV_PARSE_STYLE DISABLE
```

or

```
$ DEASSIGN DECC$ARGV_PARSE_STYLE
```

The value of the DECC$ARGV_PARSE_STYLE logical is case-insensitive.

5.4.5 Compaq C RTL Can Implicitly Open Files for Shared Access

The Compaq C RTL was enhanced to open all files for shared access as if the "shr=del,get,put,upd" option was specified in the `open*` or `creat` call.

To enable this feature, define the logical name DECC$FILE_SHARING to the value "ENABLE". The value is case-insensitive.

DECC$FILE_SHARING is checked only once per image activation, not on a file-by-file basis.

5.4.6 Alternate Way of Translating UNIX File Specifications

The Compaq C RTL was enhanced to allow interpreting the leading part of a UNIX-style file specification as either a subdirectory name or a device name.

The default translation of a "foo/bar" UNIX-style name to a "foo:bar" VMS-style name remains the default.

To translate a "foo/bar" UNIX-style name to a "[.foo]bar" VMS-style name, define the logical name DECC$DISABLE_TO_VMS_LOGNAME_TRANSLATION to ENABLE.

DECC$DISABLE_TO_VMS_LOGNAME_TRANSLATION is checked only once per image activation, not on a file-by-file basis.

5.4.7 New Functions

The Compaq C RTL has added the following functions in OpenVMS Version 7.3:

```
fchown
link
utime
utimes
writev
```

5.5 Fortran Support for 64-Bit Address (Alpha)

Support has been added to OpenVMS Alpha to allow Fortran developers to use static data in 64-bit address space.

For more information about how to use this feature, refer to the Fortran documentation.

5.6 Large Page-File Sections (Alpha)

Page-file sections are used to store temporary data in private or global (shared) sections of memory. In previous releases of OpenVMS Alpha, the maximum amount of data that could be backed up to page files was 32 GB per process (4 process page files, each 8 GB) and 504 GB per system (63 page files, each 8 GB).

With OpenVMS Alpha Version 7.3, the previous limits for page-file sections were extended significantly to take advantage of larger physical memory. Now images that use 64-bit addressing can map and access an amount of dynamic virtual memory that is larger than the amount of physical memory available on the system.

With the new design, if a process requires additional page-file space, page files can be allocated dynamically. Space is no longer reserved in a distinct page file, and pages are no longer bound to an initially assigned page file. Instead, if modified pages must be written back, they are written to the best available page file.

Each page or swap file can hold approximately 16 million pages (128 GB), and up to 254 page or swap files can be installed. Files larger than 128 GB are installed as multiple files.

Note the following DCL command display changes and system parameter changes as a result of the larger page-file section design:

- The SHOW MEMORY/FILES display reflects the nonreservable design. For example:

```
$ SHOW MEMORY/FILES

        System Memory Resources on 22-MAY-2000 19:04:19.67
Swap File Usage (8KB pages):                    Index ❶     Free        Size

DISK$ALPHASYS:[SYS48.SYSEXE]SWAPFILE.SYS
                                                  1          904         904
DISK$SWAP:[SYS48.SYSEXE]SWAPFILE.SYS;1
                                                  2         1048        1048
  Total size of all swap files:                                          1952

Paging File Usage (8KB pages):                  Index ❷     Free        Size

DISK$PAGE:[SYS48.SYSEXE]PAGEFILE.SYS;1
                                                253        16888       16888
DISK$ALPHASYS:[SYS48.SYSEXE]PAGEFILE.SYS
                                                254        16888       16888

  Total size of all paging files:                                       33776
  Total committed paging file usage: ❸                                   1964
```

❶ Number of swap files. Begins with an index value of 1 and increases in count.

❷ Number of page files. Begins with an index value of 254 and decreases in count.

❸ Total committed page file usage. As in previous releases, more pages can reside in page-file sections systemwide than would fit into installed page files.

- The SHOW MEMORY/FILES/FULL display no longer contains separate usage information for page and swap files. Because page-file information is no longer reserved, the system does not need to maintain the number of processes interested in a distinct page or swap file. For example:

```
$ SHOW MEMORY/FILES/FULL

        System Memory Resources on 22-MAY-2000 18:47:10.21
Swap File Usage (8KB pages):                    Index       Free        Size
DISK$ALPHASYS:[SYS48.SYSEXE]SWAPFILE.SYS
                                                  1          904         904

Paging File Usage (8KB pages):                  Index       Free        Size
DISK$ALPHASYS:[SYS48.SYSEXE]PAGEFILE.SYS
                                                254        16888       16888
  Total committed paging file usage:                                     1960
```

- System parameters PAGFILCNT and SWPFILCNT are now obsolete. Up to 254 page and swap files can be installed.

5.7 Multipath System Services

The new Multipath system services provide the capability to return path information and allow you to enable, disable, and switch specific I/O paths to any device.

The concept of multiple I/O paths to storage devices was introduced in OpenVMS Version 7.2-1. It is now possible to select more than one I/O path to a device in the event that the path in use should fail.

To assist in decision making when configuring a system's I/O structure, the following DCL commands were made available to allow you to display I/O path information and change the current settings affecting these paths:

- SET DEVICE device-name/PATH=path-description-string/SWITCH

- SET DEVICE device-name/PATH=path-description-string/[NO]ENABLE
- SHOW DEVICE/MULTIPATH device-name

In OpenVMS Version 7.3, the capability to return path information and allow you to enable, disable, and switch specific I/O paths to any device is now implemented in the following new system services:

- SYS$DEVICE_PATH_SCAN

  This service returns path information for a given Multipath I/O device. Each call to the service returns the name of one of the paths to the device. A **context** argument is used to maintain continuity between calls. This mechanism is similar to the one currently used for SYS$GETDVI.

- SYS$SET_DEVICE[W]

  Use this service to switch the selected path that handles I/O to a device, or to enable or disable a path for future use in the event of failover. When switching a path, the path change is initiated at the time the request is made by the system service.

  The current functions of this service include forcing an immediate path switch and enabling or disabling paths.

  A synchronous version of this service, SYS$SET_DEVICEW, is also provided. This service returns to the caller only after the path switch attempt has been made. Should the path switch fail, an error condition is returned to the caller.

  Currently, $SET_DEVICE allows only one valid item list entry.

For additional information, refer to the *OpenVMS System Services Reference Manual*.

5.8 Multiprocess Debugging (Alpha)

For Version 7.3, debugger support for multiprocess programs has been extensively overhauled. Problems have been corrected and the user interface has been improved.

The multiprocess debugging enhancements include the following features:

- Greater control over individual process and groups of processes, including:

  Execution of processes (or groups of processes)
  Suspension of processes (or groups of processes)
  Exiting processes (or groups of processes), with or without exit handler execution

- Ability to create user-defined groups of processes
- Easier to start a multiprocess debugging session; the default configuration of the kept debugger is for a multiprocess session
- Applications that use $HIBER WAIT (LIB$WAIT, $SCHDWK, and so on) can now be debugged in a multiprocess debugging session

These enhancements make it much easier to debug multiprocess programs.

5.9 Performance Application Programming Interface (API)

The Performance Application Programming Interface (API) provides a documented functional interface—the $GETRMI system service—that allows performance software engineers to access a predefined list of performance data items.

For more information about $GETRMI, refer to the *OpenVMS System Services Manual*.

5.10 POLYCENTER Software Installation Utility Enhancements

Table 5–1 shows the changes made to the product description language (PDL) for the POLYCENTER Software Installation utility.

Table 5–1 PDL Changes

Statement	Description
execute upgrade	New statement.
execute postinstall	Modified to execute on a reconfigure operation.
file *module*	Refinements made to their conflict detection and resolution algorithms. For example, when a file from the kit contains the same non-zero generation number as the same file already installed, the file from the kit is selected to replace the file on disk. Previously, in this tie situation, the file on disk was retained to resolve the conflict.
bootstrap block *execute release* *patch image* *patch text*	Obsolete. However, the utility will continue to process these statements in a backward compatible manner to support existing kits that might have used them.

Function	Description
upgrade	Enhanced to fully support version range checking.

The *POLYCENTER Software Installation Utility Developer's Guide* has been extensively revised for this release. Major improvements include:

- Updated descriptions for most PDL statements.
- A comprehensive presentation on using custom command procedures with *execute* statements (added to the Advanced Topics chapter).
- New tables, diagrams, and examples.

5.11 New Process Dump Tools (Alpha)

OpenVMS Version 7.3 contains new tools for processing dump files. Note that these new-style process dump and process dump analysis tools are not compatible with the old-style process dumps. That is, if you have a problem you want to analyze with the new tools, you must generate a new process dump using the new process dump image.

The following sections describe the new tools.

5.11.1 DCL ANALYZE/PROCESS_DUMP Command

The DCL ANALYZE/PROCESS_DUMP command invokes the OpenVMS debugger to analyze a process dump, giving you access to debugger commands for your analysis. In OpenVMS Version 7.3, most of the old DCL ANALYZE/PROCESS_ DUMP qualifiers have no effect. Only the /FULL and /IMAGE qualifiers are still valid. Both these qualifiers are still optional.

/FULL now causes the debugger to execute the debugger SHOW IMAGE, SHOW CALL, and SHOW THREAD/ALL commands after a process dump file has been opened.

/IMAGE has been renamed to /IMAGE_PATH, and is now a directory specification, rather than a file specification. /IMPAGE_PATH specifies a directory in which to look for the debug symbol information files (.DSF or .EXE files, in that order) that belong to the process dump file. The name of the symbol file *must* be the same as the image name in the process dump file. For example, for MYIMAGE.DMP, the debugger searches for file MYIMAGE.DSF or MYIMAGE.EXE.

Version 7.3 and later debuggers check for dumpfile image specification and DST file link date-time mismatches and issue a warning if one is discovered.

For more information about the DCL ANALYZE/PROCESS_DUMP command, refer to the *OpenVMS DCL Dictionary: A–M*.

5.11.2 Debugger ANALYZE/PROCESS_DUMP Command

The debugger has a new command:

ANALYZE/PROCESS_DUMP/IMAGE_PATH[=directory-spec] dumpfile.

This command is available only in the kept debugger. The kept debugger is the image you invoke with the command DEBUG/KEEP, which allows you to run and rerun programs from the same debugging session.

The qualifier /PROCESS_DUMP is required.

For more information, refer to the *OpenVMS Debugger Manual*.

5.11.3 Debugger SDA Command

The new debugger SDA command invokes the System Dump Analyzer (SDA) to allow you to look at a process dump from within the OpenVMS debugger. For example:

```
DBG> SDA

OpenVMS (TM) Alpha process dump analyzer

SDA> ..
.
.
SDA> EXIT
DBG>
```

This allows you to use SDA to analyze a process dump without terminating a debugger session.

For more information, refer to the *OpenVMS Debugger Manual*.

5.11.4 Analyzing Process Dumps on Different Systems

You can analyze a process dump file on a system different from the one on which it was generated. However, if there is a base image link date/time mismatch between the generating system and analyzing system, you must copy SYS$BASE_IMAGE.EXE from the generating system and point to it with the SDA$READ_DIR logical name. For example:

```
$ COPY other_node::SYS$LOADABLE_IMAGES:SYS$BASE_IMAGE.EXE my_disk$:[my_dir]
$ DEFINE/USER SDA$READ_DIR my_disk$:[my_dir],SYS$SYSROOT:[SYS$LDR],SYS$SYSROOT:[SYSLIB]
$ ANALYZE/PROCESS_DUMP mycrash.dmp
```

For threaded process dump analysis on a system different from the one on which it was generated, it may also be necessary to copy and logically point to the generating system's PTHREAD$RTL and PTHREAD$DBGSHR (POSIX Threads Library debug assistant). For example:

```
$ COPY other_node::SYS$LOADABLE_IMAGES:SYS$BASE_IMAGE.EXE my_disk$:[my_dir]
$ COPY other_node::SYS$SHARE:PTHREAD$RTL.EXE my_disk$:[my_dir]
$ COPY other_node::SYS$SHARE:PTHREAD$DBGSHR.EXE my_disk$:[my_dir]
$ DEFINE/USER SDA$READ_DIR my_disk$:[my_dir],SYS$SYSROOT:[SYS$LDR],SYS$SYSROOT:[SYSLIB]
$ DEFINE/USER PTHREAD$RTL my_disk$:[my_dir]PTHREAD$RTL.EXE
$ DEFINE/USER PTHREAD$DBGSHR my_disk$:[my_dir]PTHREAD$DBGSHR.EXE
$ ANALYZE/PROCESS_DUMP mycrash.dmp
```

5.11.5 Forcing a Process Dump

You can force a process dump with the DCL command SET PROCESS/DUMP=NOW process-spec. This command causes the contents of the address space occupied by process-spec to be written immediately to the file named image-name.DMP in the current directory (image-name is the same as the file name).

For more information about the DCL SET PROCESS/DUMP command, refer to the *OpenVMS DCL Dictionary: N–Z*.

5.11.6 Process Dumps: Security and Diskquota Guidelines

A process dump is either complete or partial. A **complete process dump** contains all of process space and all process-pertinent data from system space. A **partial process dump** contains only user-readable data from process space and only those data structures from system space that are not deemed sensitive. Privileged or protected data, such as an encryption key in third-party software, might be considered sensitive.

In general, nonprivileged users should not be able to read complete process dumps, and by default they cannot do so. However, certain situations require nonprivileged users to be able to read complete process dumps. Other situations require enabling a user to create a complete process dump while at the same time preventing that user from being able to read the complete process dump.

By default, process dumps are written to the current default directory of the user. The user can override this by defining the logical name SYS$PROCDMP to identify an alternate directory path. Note that the name of the process dump file is always the same as the name of the main image at the time the process dump is written, with the file extension .DMP.

5.11.6.1 Special Rights Identifiers

You can use the new rights identifier IMGDMP$READALL to allow a nonprivileged user to read a complete process dump. You can use the new rights identifier IMGDMP$PROTECT to protect a complete process dump from being read by the user that created the process dump. These rights identifiers are created during the installation of OpenVMS Version 7.3 by the image SYS$SYSTEM:IMGDMP_RIGHTS.EXE, which is also run automatically during system startup to ensure that these rights identifiers exist with the correct values and attributes.

If these rights identifiers have been deleted, you can run SYS$SYSTEM:IMGDMP_RIGHTS.EXE to recreate them. For example:

```
$ RUN SYS$SYSTEM:IMGDMP_RIGHTS
%PROCDUMP-I-CREATED, rights identifier IMGDMP$READALL successfully created
%PROCDUMP-I-CREATED, rights identifier IMGDMP$PROTECT successfully created
```

Note that IMGDMP$READALL has no attributes, but IMGDMP$PROTECT is created with the RESOURCE attribute.

5.11.6.2 Privileged Users and Process Dumps

For this discussion, a privileged user is one who satisfies one of the following conditions:

- Has one or more of the privileges CMKRNL, CMEXEC, SYSPRV, READALL, or BYPASS
- Is a member of a system UIC group (by default [10,n] or lower). Such users are treated as though they hold SYSPRV privilege.

Holders of CMKRNL or CMEXEC can write complete process dumps. Holders of any of the other privileges can read a process dump wherever it has been written.

5.11.6.3 Nonprivileged Users and Process Dumps

To allow a nonprivileged user to write and read complete process dumps, grant the rights identifier IMGDMP$READALL to the user. If the IMGDMP$READALL rights identifier does not exist, run the image SYS$SYSTEM:IMGDMP_RIGHTS.EXE to create it (see Section 5.11.6.1). Then use AUTHORIZE to grant the rights identifier to the user. For example:

```
$ DEFINE /USER SYSUAF SYS$SYSTEM:SYSUAF.DAT  !if necessary
$ RUN SYS$SYSTEM:AUTHORIZE
UAF> GRANT /IDENTIFIER IMGDMP$READALL <user>
UAF> EXIT
```

Note that the user must log out and log in again to be able to receive the rights identifier. A nonprivileged user with rights identifier IMGDMP$READALL can read and write complete process dumps without restriction.

5.11.6.4 Protecting Process Dumps

You can allow a nonprivileged user to write a complete process dump and at the same time prevent the user from reading the process dump just written. To do so, perform the following procedure:

1. If the IMGDMP$PROTECT rights identifier does not exist, run the image SYS$SYSTEM:IMGDMP_RIGHTS.EXE to create it (see Section 5.11.6.1).

2. Create a protected directory with rights identifier IMGDMP$PROTECT. For example:

```
$ CREATE /DIRECTORY DKA300:[PROCDUMPS] -
   /PROTECTION=(S:RWE,O:RWE,G,W) /OWNER_UIC=IMGDMP$PROTECT
$ SET SECURITY DKA300:[000000]PROCDUMPS.DIR -
   /ACL=((DEFAULT_PROTECTION,SYSTEM:RWED,OWNER:RWED,GROUP:,WORLD:), -
  (IDENTIFIER=IMGDMP$PROTECT,ACCESS=READ+WRITE), -
  (IDENTIFIER=IMGDMP$PROTECT,OPTIONS=DEFAULT, -
   ACCESS=READ+WRITE+EXECUTE+DELETE+CONTROL), -
  (CREATOR,ACCESS=NONE))
```

3. Define protected logical name SYS$PROTECTED_PROCDMP to point to the protected directory. For example:

```
$ DEFINE /SYSTEM /EXECUTIVE_MODE SYS$PROTECTED_PROCDMP DKA300:[PROCDUMPS]
```

   If DISKQUOTA is to be used on the disk containing the protected directory, specify the maximum disk space to be used for process dumps. For example:

```
$ RUN SYS$SYSTEM:SYSMAN
SYSMAN> DISKQUOTA CREATE /DEVICE=DKA300  ! if necessary
SYSMAN> DISKQUOTA ENABLE /DEVICE=DKA300  ! if necessary
SYSMAN> DISKQUOTA ADD IMGDMP$PROTECT /DEVICE=DKA300 /PERMQUOTA=10000
SYSMAN> DISKQUOTA REBUILD /DEVICE=DKA300 ! if necessary
SYSMAN> EXIT
```

Caution

Do not grant IMGDMP$PROTECT to any user. It is granted and revoked as needed by SYS$SHARE:IMGDMP.EXE from executive mode while writing a process dump. If you grant it permanently to a user, that user has access to all process dumps written to the protected directory.

You can choose to set up additional ACLs on the protected directory to further control which users are allowed to read and write process dumps there.

Note that to take a process dump when the image is installed with elevated privileges or belongs to a protected subsystem, the user must hold CMKRNL privilege, and is by definition a privileged user (see Section 5.11.6.2).

5.12 RMS Locking Enhancements

This section introduces the new Record Management Services (RMS) enhancements provided in this release.

5.12.1 RMS Locking Performance Enhancements (Alpha)

The following sections describe RMS locking performance enhancements that are in OpenVMS Alpha Version 7.2-1H1 and in OpenVMS Version 7.3.

5.12.1.1 RMS Global Buffer Read-Mode Locking

In the RMS run-time processing environment, the use of global buffers can minimize I/O operations for shared files. This release introduces read-mode bucket locking that minimizes locking for shared access to global buffers. This new functionality:

- Allows concurrent read access to the global buffers. Accesses are no longer serialized, waiting to acquire an exclusive lock for a read access.

- Caches the read-mode lock as a system lock, which is retained over accesses and only lowered to null when the lock is blocking an exclusive write request. This functionality significantly reduces both local and remote lock request traffic (the number of $ENQ and $DEQ system service calls) as well as associated IPL-8 spinlock activity and System Communications Services (SCS) messages for a cluster.
- Does not increase lock resource names or the number of active system or process locks on the system.
- Is functionally compatible in mixed version clusters that include both Alpha and VAX computers.

This new functionality applies to read operations (using the $GET and $FIND services) for all three file organizations: sequential, relative, and indexed. It also applies to a write operation (using the $PUT service) for the read accesses used for index buckets the first time through an index tree for the write.

You do not need to make changes to existing applications to implement the read-only global bucket locks. However, global buffers must be set on a data file to take advantage of the enhancement. Use the following DCL command, where *n* is the number of buffers:

```
$ SET FILE/GLOBAL_BUFFER=n <filename>
```

For information about specifying the number of buffers, refer to the *OpenVMS DCL Dictionary*. For general information about using global buffers, refer to the section entitled Using Global Buffers for Shared Files in the *Guide to OpenVMS File Applications*.

In a mixed cluster environment where there may be high contention for specific buckets, the Alpha nodes that are using read-mode global bucket locking may dominate accesses to write-shared files, thereby preventing timely access by other nodes.

With the new /CONTENTION_POLICY=*keyword* qualifier to the SET RMS_DEFAULT command, you can specify the level of locking fairness at the process or system level for environments that experience high contention conditions.

For more information about using the /CONTENTION_POLICY=*keyword* qualifier, refer to the SET RMS_DEFAULT section of the *OpenVMS DCL Dictionary*.

5.12.1.2 No Query Record Locking Option

This release introduces new functionality that can minimize record locking for read accesses to shared files, thereby avoiding the processing associated with record locking calls to the Lock Manager.

In previous releases, if a file is opened allowing write sharing, an exclusive record lock is taken out for all record operations (both read and write). Applications may obtain record locking modes other than the exclusive lock (default) by specifying certain options to the RAB$L_ROP field. However, all the options involve some level of record locking. That is, the options require $ENQ or $DEQ system service calls to the Lock Manager.

The user record locking options include the RAB$V_NLK (no lock) query locking option, which requests that RMS take out a lock to probe for status and not hold the lock for synchronization. If the lock is not granted (exclusive lock held) and the read-regardless (RAB$V_RRL) option is not set, the record access fails with

an RMS$_RLK status. Otherwise, the record is returned with one of the following statuses:

- RMS$_SUC — No other writers
- RMS$_OK_RLK — Record can be read but not written
- RMS$_OK_RRL — Exclusive lock is held (lock request denied) but the read-regardless (RAB$V_RRL) option is set

When only the RAB$V_NLK option is specified, record access can be denied. When both the RAB$V_NLK and RAB$V_RRL options are specified, an application can guarantee the return of any record with a success or alternate success status.

This release introduces the **no query record locking option**, which allows applications to read records (using $GET or $FIND services) without any consideration of record locking. This option:

- Does not make a call to the Lock Manager
- Is equivalent to both RAB$V_NLK and RAB$V_RRL being set except that the RMS$_OK_RLK or RMS$_OK_RRL status will not be returned

This functionality is independent of bucket locks. It applies to both local and global buffers and to all three file organizations (sequential, relative, and indexed).

Three alternate methods for specifying the no query record locking option are outlined in Table 5–2.

Note the following:

- The first method allows the option to be enabled externally, potentially without any application change.
- You should use any of the methods only as appropriate for the application. In particular, you should check for any dependency in an existing application on the alternate success status RMS$_OK_RLK or RMS$_OK_RRL.

Table 5–2 Methods Available for Specifying No Query Record Locking

To...	Use This Method...
Disable query record locking at the process or system level.	Enter the following DCL command to request that RMS use no query record locking for any read operation with both RAB$V_NLK and RAB$V_RRL options set in the RAB$L_ROP field: `$ SET RMS_DEFAULT/QUERY_LOCK=DISABLE[/SYSTEM]` Keys on RAB$V_NLK and RAB$V_RRL options in existing applications.

(continued on next page)

Table 5–2 (Cont.) Methods Available for Specifying No Query Record Locking

To...	Use This Method...
Enable no query record locking on a per-record read operation.	Set the RAB$V_NQL option in the RAB$W_ROP_2 field. The RAB$V_NQL option takes precedence over all other record locking options. Use only if the current read ($GET or $FIND) operation is not followed by an $UPDATE or $DELETE call.
Enable no query record locking at the file level.	Set the FAB$V_NQL option in the FAB$B_SHR field to request that RMS use no query locking for the entire period the file is open for any read record operation with both RAB$V_NLK and RAB$V_RRL options set in the RAB$L_ROP field. This option can be used with any combination of the other available FAB$B_SHR sharing options. Keys on RAB$V_NLK and RAB$V_RRL options in applications.

RMS precedence for the no query record locking option is as follows:

- The RAB$V_NQL option set in the RAB$W_ROP_2 field
- At file open (and applied, if RAB$V_NLK and RAB$V_RRL are set for the read operation):
  - The FAB$V_NQL option set in the FAB$B_SHR field
  - The SET RMS_DEFAULT/QUERY_LOCK=DISABLE setting at the process level
  - The SET RMS_DEFAULT/QUERY_LOCK=DISABLE setting at the system level. If the process /QUERY_LOCK setting equals SYSTEM_DEFAULT (the default when the process is created), RMS uses the system specified value.

For more information, see *OpenVMS Record Management Services Reference Manual*.

5.12.2 Record Locking Options to Control Deadlock Detection

RMS uses the distributed Lock Manager ($ENQ system service) for record locking.

To help prevent false deadlocks, the distributed Lock Manager uses the following flags for lock requests:

Flag[1]	Purpose
LCK$M_NODLCKWT	When set, the lock management services do not consider this lock when trying to detect deadlock conditions.
LCK$M_NODLCKBLK	When set, the lock management services do not consider this lock as blocking other locks when trying to detect deadlock conditions.

[1]Improper use of these flags can result in the lock management services ignoring genuine deadlocks. For complete flag information, refer to the $ENQ section of the *OpenVMS System Services Reference Manual: A–GETUAI*.

In previous releases, RMS did not set these flags in its record lock requests.

With this release, you can optionally request that RMS set these flags in record lock requests by setting the corresponding options RAB$V_NODLCKWT and RAB$V_NODLCKBLK in the new RAB$W_ROP_2 field. For more information about using these options, refer to the flag information in the $ENQ section of the *OpenVMS System Services Reference Manual: A–GETUAI*.

5.13 OpenVMS Registry

Beginning in OpenVMS Version 7.3, the $REGISTRY system service and the OpenVMS Registry server have been enhanced to use the Intra-Cluster Communications (ICC) protocol. ICC provides a high-performance communication mechanism that is ideal for large transfers. Using ICC eases restrictions on the amount of data that can be transferred between the $REGISTRY system service and the Registry server. These restrictions previously prevented large key values from being stored and retrieved, and prevented full searches of large databases. The changes made in OpenVMS Version 7.3 result in an incompatibility between the OpenVMS Version 7.2 $REGISTRY service and Registry server and the OpenVMS Version 7.3 $REGISTRY service and Registry server. However, these changes substantially benefit OpenVMS customers in this release and in future releases, when we plan to further reduce these restrictions.

Also in OpenVMS Version 7.3, registry operations are client/server based, and as such require some length of time for the server to respond to a request. If the server is too busy or the timeout value is too small, or both, the server will not respond in time and the $REGISTRY service will return a REG$_NORESPONSE error. This does not necessarily mean that the operation failed; it only means that the server was not able to respond before the time expired. Most operations complete immediately. However, Compaq recommends that you specify the timeout value be a minimum of 5 seconds.

The new format of the $registry system service is:

```
$REGISTRY [efn], func, [ntcredentials], itmlst, [iosb] [,astadr]
[,astprm] [,timeout]
```

Note that `astadr`, `astprm` and `timeout` are optional arguments. These optional arguments cannot be defaulted, which means that to specify the `timeout` argument, you must specify `astadr` and `astprm` (or specify them as 0). Some languages, such as Bliss and Macro, provide macros to do this for you.

5.13.1 REG$CP Registry Utility

The REG$CP Registry Utility has been enhanced to use the `timeout` argument. REG$CP commands now support a `/WAIT=numberofseconds` qualifier, allowing you to specify the number of seconds to wait for the Registry Server to respond to the command. `/WAIT` is negatable (by using `/NOWAIT`). However, like the `timeout` argument, Compaq recommends that you specify a minimum of 5 seconds.

The REG$CP Registry Utility has also been enhanced to display security descriptors. The LIST command can now be used to display the security descriptor associated with a particular key. This includes the security descriptor structure itself, and may also include Security Identifiers (SIDs), System Access-Control Lists (SACLs), and Discretionary Access-Control Lists (DACLs). You must have access to the key to display the security descriptor; in other words, you must have proper credentials to read the security information, or you must be suitably privileged.

For more information, refer to the *OpenVMS Connectivity Developer Guide*, which is available on the OpenVMS Alpha CD-ROM in directory [COM_ALPHA_011A].

5.14 Alpha SDA Commands, Parameters, and Qualifiers

The OpenVMS Version 7.3 software release offers a number of new Alpha SDA commands, parameters, and qualifiers. OpenVMS Version 7.3 also offers many new parameters and qualifiers for existing commands.

For more detailed information, refer to the *OpenVMS Alpha System Analysis Tools Manual*.

5.14.1 New Alpha SDA Commands

The following section lists and defines the new System Dump Analyzer commands with their parameters and qualifiers.

5.14.1.1 DUMP

The DUMP command displays the contents of a range of memory formatted as a comma-separated variable (CSV) list, suitable for inclusion in a spreadsheet.

The following table shows the parameter for the DUMP command:

Parameter	Meaning
range	The range of locations to be displayed. The range is specified in one of the following formats:

Format	Meaning
m:n	Range from address *m* to address *n* inclusive
m;n	Range from address *m* for *n* bytes

The following table shows the qualifiers for the DUMP command:

Qualifier	Meaning
/COUNT=[{ALL|records}]	Gives the number of records to be displayed. The default is to display all records.
/DECIMAL	Outputs data as decimal values.
/FORWARD	Causes SDA to display the records in the history buffer in ascending address order. This is the default.
/HEXADECIMAL	Outputs data as hexadecimal values. This is the default.
/INDEX_ARRAY [={LONGWORD (default)|QUADWORD}]	Indicates to SDA that the range of addresses given is a vector of pointers to the records to be displayed. The vector can be a list of longwords (default) or quadwords. The size of the range must be an exact number of longwords or quadwords as appropriate.
/INITIAL_POSITION ={ADDRESS=address|RECORD=number}	Indicates to SDA which record is to be displayed first. The default is the lowest addressed record if /FORWARD is used, and the highest addressed record if /REVERSE is used. The initial position may be given as a record number within the range, or the address at which the record is located.
/LONGWORD	Outputs each data item as a longword. This is the default.
/PHYSICAL	Indicates to SDA that all addresses (range and/or start position) are physical addresses. By default, virtual addresses are assumed.
/QUADWORD	Outputs each data item as a quadword.
/RECORD_SIZE=size	Indicates the size of each record within the history buffer, the default being 512 bytes. Note that this size must exactly divide into the total size of the address range to be displayed, unless /INDEX_ARRAY is specified.
/REVERSE	Causes SDA to display the records in the history buffer in descending address order.

5.14.1.2 SET SYMBOLIZE

The SET SYMBOLIZE command enables or disables symbolization of addresses in the display from an EXAMINE command.

The following shows the parameters for the SET SYMBOLIZE command:

Parameter	Meaning
ON	Enables symbolization of addresses
OFF	Disables symbolization of addresses

There are no qualifiers for this command.

5.14.1.3 SHOW MEMORY

The SHOW MEMORY command displays the availability and usage of those memory resources that are related to memory.

There are no parameters for this command. The following shows the qualifiers for the SHOW MEMORY command, which are the same as for the existing DCL command:

Qualifier	Meaning
/ALL	Displays all available information; that is, information displayed by the /FILES, /PHYSICAL_PAGES, /POOL, and /SLOTS qualifiers. This is the default display.
/BUFFER_OBJECTS	Displays information about system resources used by buffer objects.
/CACHE	Displays information about the Virtual I/O Cache facility. The cache facility information is displayed as part of the SHOW MEMORY and SHOW MEMORY/CACHE/FULL commands.
/FILES	Displays information about the use of each paging and swapping file currently installed.
/FULL	Displays additional information about each pool area or paging or swapping file currently installed, when used with the /POOL or the /FILES qualifier. This qualifier is ignored unless the /FILES or the /POOL qualifier is specified explicitly. When used with the /CACHE qualifier, /FULL displays additional information about the use of the Virtual I/O Cache facility.
/GH_REGIONS	Displays information about the granularity hint regions (GHR) that have been established. For each of these regions, information is displayed about the size of the region, the amount of free memory, the amount of memory in use, and the amount of memory released to OpenVMS from the region. The granularity hint regions information is also displayed as part of SHOW MEMORY, SHOW MEMORY/ALL, and SHOW MEMORY/FULL commands.
/PHYSICAL_PAGES	Displays information about the amount of physical memory and the number of free and modified pages.
/POOL	Displays information about the usage of each dynamic memory (pool) area, including the amount of free space and the size of the largest contiguous block in each area.
/RESERVED	Displays information about memory reservations.
/SLOTS	Displays information about the availability of partition control block (PCB) vector slots and balance slots.

5.14.1.4 SHOW RAD

The SHOW RAD command displays the settings and explanations of the RAD_SUPPORT system parameter fields, and the assignment of CPUs and memory to the Resource Affinity Domains (RADs). This command is only useful on platforms that support RADs. By default, the SHOW RAD command displays the settings of the RAD_SUPPORT system parameter fields.

The following shows the parameter for the SHOW RAD command:

Parameter	Meaning
number	Displays information on CPUs and memory for the specified RAD

The following shows the qualifier for the SHOW RAD command:

Qualifier	Meaning
/ALL	Displays settings of the RAD_SUPPORT parameter fields and the CPU and memory assignments for all RADs

5.14.1.5 SHOW TQE

The SHOW TQE command displays the entries in the Timer Queue. The default output is a summary display of all timer queue entries (TQEs) in chronological order.

There are no parameters for this command. The following shows the qualifiers for the SHOW TQE command:

Qualifier	Meaning
/ADDRESS=*n*	Outputs a detailed display of the TQE at the specified address
/ALL	Outputs a detailed display of all TQEs
/BACKLINK	Outputs the display of TQEs, either detailed (/ALL) or brief (default), in reverse order, starting at the entry furthest into the future
/PID=*n*	Limits the display of the TQEs that affect the process with the specified *internal* PID
/ROUTINE=*n*	Limits the display of the TQEs for which the specified address is the fork PC

5.14.1.6 UNDEFINE

The UNDEFINE command causes SDA to remove the specified symbol from its symbol table.

The following shows the parameter for the UNDEFINE command:

Parameter	Meaning
symbol	The name of the symbol to be deleted from SDA's symbol table. A symbol name is required.

There are no qualifiers for this command.

5.14.2 New Parameters and Qualifiers for Existing Commands

The following section lists and defines new parameters and qualifiers for existing commands.

5.14.2.1 REPEAT

The REPEAT command has the following new parameter:

Parameter	Meaning
count	The number of times the previous command is to be repeated. The default is a single repeat.

The REPEAT command has the following new qualifier:

Qualifier	Meaning
/UNTIL=condition	Defines a condition that terminates the REPEAT command. By default, there is no terminating condition.

5.14.2.2 SEARCH

The /STEPS qualifier of the SEARCH command now allows any step size. In addition to the keywords QUADWORD, LONGWORD (default), WORD, or BYTE, any value can be specified.

Qualifier	Meaning
/STEPS={QUADWORD|LONGWORD|WORD |BYTE|*value*}	Specifies the step factor of the search through the specified memory **range**. After the SEARCH command has performed the comparison between the value of **expression** and memory location, it adds the specified step factor to the address of the memory location. The resulting location is the next location to undergo the comparison. If you do not specify the /STEPS qualifier, the SEARCH command uses a step factor of a longword.

5.14.2.3 SET OUTPUT

The SET OUTPUT command has the following new qualifiers:

Qualifier	Meaning
/[NO]HEADER	The /HEADER qualifier causes SDA to include a heading at the top of each page of the output file. This is the default. The /NOHEADER qualifier causes SDA to omit the page headings. Use of /NOHEADER implies /NOINDEX.
/SINGLE_COMMAND	Indicates to SDA that the output for a single command is to be written to the specified file and that subsequent output should be written to the terminal.

5.14.2.4 SET PROCESS

The SET PROCESS command has the following new qualifier:

Qualifier	Meaning
/NEXT	Causes SDA to locate the next valid process in the process list and select that process. If there are no further valid processes in the process list, SDA returns an error.

5.14.2.5 SHOW DEVICE

The SHOW DEVICE command has the following new qualifiers:

Qualifier	Meaning
/CDT=*address*	Identifies the device by the address of its Connector Descriptor Table (CDT). This applies to cluster port devices only.
/PDT	Displays the Memory Channel Port Descriptor Table. This qualifier is ignored for devices other than memory channel.
/UCB=*ucb-address*	This is a synonym for /ADDRESS=*ucb-address*.

5.14.2.6 SHOW GCT

The SHOW GCT command has the following new qualifier:

Qualifier	Meaning
/CHILDREN	When used with /ADDRESS=*n* or /HANDLE=*n*, the /CHILDREN qualifier causes SDA to display all nodes in the configuration tree that are children of the specified node.

5.14.2.7 SHOW LOCK

The SHOW LOCK command's qualifier /STATUS has the following new keyword:

Keyword	Meaning
DPC	Indicates a delete pending cache lock

5.14.2.8 SHOW PFN_DATA

The SHOW PFN_DATA command has the following new qualifier:

Qualifier	Meaning
/RAD [={n|ALL}]	Displays data on the disposition of pages among the Resource Affinity Domain on applicable systems

5.14.2.9 SHOW POOL

The SHOW POOL command has the following new qualifiers:

Qualifier	Meaning
/BRIEF	Displays only general information about pool and its addresses.
/CHECK	Checks all free packets for POOLCHECK-style corruption, in exactly the same way that the system does when generating a POOLCHECK crashdump.
/MAXIMUM_BYTES [=*n*]	Displays only the first *n* bytes of a pool packet; default is 64 bytes.
/STATISTICS [= ALL]	Displays usage statistics about each lookaside list and the variable free list. For each lookaside list, its queue header address, packet size, the number of packets, attempts, fails, and deallocations are displayed. (If pool checking is disabled, the attempts, fails, and deallocations are not displayed.) For the variable free list, its queue header address, the number of packets and the size of the smallest and largest packets are displayed. /STATISTICS can be further qualified by using either /NONPAGED, /BAP, or /PAGED to display statistics for a specified pool area. (Note that for paged pool, only variable free list statistics are displayed.) If /STATISTICS is specified without the ALL keyword, only active lookaside lists are displayed. Use /STATISTICS = ALL to display all lookaside lists.
/UNUSED	Displays only variable free packets and lookaside list packets, not used packets.

5.14.2.10 SHOW PROCESS

The SHOW PROCESS command has the following new qualifiers:

Qualifier	Meaning
/FID_ONLY	When used with /CHANNEL or /PROCESS_ SECTION_TABLE (/PST), the /FID_ONLY qualifier causes SDA to not attempt to translate the FID (File ID) to a file name when invoked with ANALYZE/SYSTEM.
/GSTX=*index*	When used with the /PAGE_TABLES qualifier, it causes SDA to only display page table entries for the specific global section.
/IMAGES [= ALL]	By default, /IMAGES now only displays the address of the image control block, the start and end addresses of the image, the activation code, the protected and shareable flags, the image name, and the major and minor IDs of the image. If /IMAGES = ALL qualifier is used, it also displays the base, end, image offset, and section type for installed resident images in use by this process.
/NEXT	Causes SDA to locate the next valid process in the process list and select that process. It there are no further valid processes in the process list, SDA returns an error.
/PST	This is a synonym for /PROCESS_SECTION_TABLE.

5.14.2.11 SHOW RESOURCE

The SHOW RESOURCE command has the following new qualifier:

Qualifier	Meaning
/OWNED	Causes SDA to only display owned resources

5.14.2.12 SHOW SPINLOCKS

The SHOW SPINLOCKS command has the following new qualifier:

Qualifier	Meaning
/COUNTS	Produces a display of Acquire, Spin, and Wait counts for each spinlock

5.14.2.13 SHOW SUMMARY

The SHOW SUMMARY command has the following new qualifier:

Qualifier	Meaning
/PROCESS_NAME=*process_name*	Displays only processes with the specified process name. Wildcards can be used in *process_name*, in which case SDA displays all matching processes. The default action is for SDA to display data for all processes, regardless of process name.

5.15 New SDA Commands for the Spinlock Tracing Utility

The OpenVMS Version 7.3 software release includes the new Spinlock Tracing utility. With the implementation of this utility, you can now tell which spinlock is heavily used, and who is acquiring and releasing the contended spinlocks. The Spinlock Tracing utility allows a characterization of spinlock usage, as well as collection of performance data for a given spinlock on a per-CPU basis. The tracing ability can be enabled or disabled while the system is running, allowing the collection of spinlock data for a given period of time without system interruption.

To use the Spinlock Tracing utility, SDA has implemented new commands and qualifiers. These SDA commands and qualifiers are described as follows:

5.15.1 SPL LOAD

This command loads the SPL$DEBUG execlet. This must be done prior to starting spinlock tracing. It has no qualifiers.

5.15.2 SPL SHOW COLLECT

This command displays the data collected for a specific spinlock. It has no qualifiers.

5.15.3 SPL SHOW TRACE

This command displays spinlock tracing information. Table 5–3 shows the qualifiers for this command.

Table 5–3 Qualifiers for the SPL SHOW TRACE Command

Qualifier	Meaning
/SPINLOCK=*spinlock*	Specifies the display of a specific spinlock, for example, /SPINLOCK=LCKMGR or /SPINLOCK=SCHED.
/NOSPINLOCK	Specifies that no spinlock trace information be displayed. If omitted, all spinlock trace entries are decoded and displayed.
/FORKLOCK=*forklock*	Specifies the display of a specific forklock, for example, /FORKLOCK=IOLOCK8 or /FORKLOCK=IPL8.
/NOFORKLOCK	Specifies that no forklock trace information be displayed. If omitted, all fork trace entries are decoded and displayed.
/ACQUIRE	Displays any spinlock acquisitions.
/NOACQUIRE	Ignores any spinlock acquisitions.
/RELEASE	Displays any spinlock releases.
/NORELEASE	Ignores any spinlock releases.
/WAIT	Displays any spinwait operations.
/NOWAIT	Ignores any spinwait operations.
/FRKDSPTH	Displays all invocations of fork routines within the fork dispatcher. This is the default.
/NOFRKDSPTH	Ignores all of the operations of the /FRKDSPTH qualifier.
/FRKEND	Displays all returns from fork routines within the fork dispatcher. This is the default.
/NOFRKEND	Ignores all operations of the /FRKEND qualifier.
/SUMMARY	Stops the entire trace buffer and displays a summary of all spinlock and forklock activity. It also displays the top ten callers.
/CPU=*n*	Specifies the display of information for a specific CPU only, for example, /CPU=5 or /CPU=PRIMARY. By default, all trace entries for all CPUs are displayed.
/TOP=*n*	Displays a different number other than the top ten callers or fork PC. By default, the top ten are displayed. This qualifier is only useful when you also specify the /SUMMARY qualifier.

5.15.4 SPL START COLLECT

This command accumulates information for a specific spinlock. Table 5–4 shows the qualifiers for this command:

Table 5–4 Qualifiers for the SPL START COLLECT Command

Qualifier	Meaning
/SPINLOCK=*spinlock*	Specifies the tracing of a specific spinlock, for example, /SPINLOCK=LCKMGR or /SPINLOCK=SCHED
/ADDRESS=*n*	Specifies the tracing of a specific spinlock by address

5.15.5 SPL START TRACE

This command enables spinlock tracing. Table 5–5 shows the qualifiers for this command.

Table 5–5 Qualifiers for the SPL START TRACE Command

Qualifier	Meaning
/SPINLOCK=*spinlock*	Specifies the tracing of a specific spinlock.
/NOSPINLOCK	Disables spinlock tracing and does not collect any spinlock data. If omitted, all spinlocks are traced.
/FORKLOCK=*forklock*	Specifies the tracing of a specific forklock, for example, /FORKLOCK=IOLOCK8 or /FORKLOCK=IPL8.
/NOFORKLOCK	Disables forklock tracing and does not collect any forklock data. If omitted, all forks are traced.
/BUFFER=*pages*	Specifies the size of the trace buffer (in Alpha page units). It defaults to 128 pages, which is equivalent to 1MB, if omitted.
/ACQUIRE	Traces any spinlock acquisitions. This is the default.
/NOACQUIRE	Ignores any spinlock acquisitions.
/RELEASE	Traces any spinlock releases. This is the default.
/NORELEASE	Ignores any spinlock releases.
/WAIT	Traces any spinwait operations. This is the default.
/NOWAIT	Ignores any spinwait operations.
/FRKDSPTH	Traces all invocations of fork routines within the fork dispatcher. This is the default.
/NOFRKDSPTH	Ignores all of the /FRKDSPTH operations.
/FRKEND	Traces all returns from fork routines within the fork dispatcher. This is the default.
/NOFRKEND	Ignores all of the operations of the /FRKEND qualifier.
/CPU=*n*	Specifies the tracing of a specific CPU only, for example, /CPU=5 or /CPU=PRIMARY. By default, all CPUs are traced.

5.15.6 SPL STOP COLLECT

This command stops the spinlock collection, but does not stop spinlock tracing. It has no qualifiers.

5.15.7 SPL STOP TRACE

This command disables spinlock tracing, but it does not deallocate the trace buffer. It has no qualifiers.

5.15.8 SPL UNLOAD

This command unloads and cleans up the SPL$DEBUG execlet. Tracing is automatically disabled and the trace buffer deallocated. It has no qualifiers.

For more information, refer to the *OpenVMS Alpha System Analysis Tools Manual*.

5.16 System Services

The following table describes new and updated system services for OpenVMS Version 7.3.

For additional information, refer to the *OpenVMS System Services Reference Manual*.

System Service	Documentation Update
$CHECK_PRIVILEGES	The description of the 'prvadr' argument has been updated.
$CLRAST	This service has been documented for Version 7.3.
$DCLEXH	The description has been updated, and a BASIC example has been added.
$DELETE_INTRUSION	This service has been updated in support of Clusterwide Intrusion.
$DEVICE_PATH_SCAN	This is a new service in support of Multipath.
$DISMOU	The following item codes have been added: DMT$M_MINICOPY_REQUIRED, DMT$M_MINICOPY_OPTIONAL, and DMT$M_FORCE.
$EXPREG	The text for condition value, SS$_ILLPAGCNT, has been updated.
$GETDVI	The item codes, MT3_DENSITY and MT3_SUPPORTED, have been added.
	The item codes, DVI$_FC_NODE_NAME, DVI$_FC_PORT_NAME, and DVI$_WWID, have been added.
	The description for the DVI$_MOUNTCNT item code has been updated.
$GETJPI	The following item codes have been added: JPI$_RMS_DFMBC, JPI$_RMS_DFMBFIDX, JPI$_RMS_DFMBFREL, JPI$_RMS_DFMBFSDK, JPI$_RMS_DFMBFSMT, JPI$_RMS_DFMBFSUR, JPI$_RMS_DFNBC, JPI$_RMS_EXTEND_SIZE, JPI$_RMS_FILEPROT, and JPI$_RMS_PROLOGUE.
	The following item codes have been added for Multithreads support: JPI$_INITIAL_THREAD_PID, JPI$_KT_COUNT, JPI$_MULTITHREAD, and JPI$_THREAD_INDEX.
	The code example has been updated for VAX and Alpha usage.
$GETRMI	This is a new service in support of Performance API.
$GETQUI	The item code, QUI$V_JOB_REQUEUE, has been added.
$GETSYI	The item code, SYI$_SERIAL_NUMBER, has been added.
$IO_PERFORM	The 'porint' argument in the format section has been changed to 'devdata, to match the C prototype.
$MGBLSC	The text for the 'inadr' argument has been updated, and the SS$_INVARG condition value has been added.
$MOUNT	The following item codes have been added: MNT$M_MINICOPY_OPTIONAL, MNT$M_MINICOPY_REQUIRED, MNT$M_REQUIRE_MEMBERS, and MNT$M_VERIFY_LABELS.
$PERSONA_QUERY	Tables for Common, General, and NT item codes have been added.

System Service	Documentation Update
$PROCESS_SCAN	The following item codes have been added for Multithreads support: PSCAN$_KT_COUNT and PSCAN$_ MULTITHREAD.
$REGISTRY	This service is now documented in the *OpenVMS System Services Reference Manual: GETUTC–Z* and online help.
$SCAN_INTRUSION	This service has been updated in support of Clusterwide Intrusion.
$SCHED	The condition value, SS$_INCLASS, has been added, and SS$_ILLSER has been deleted.
$SET_DEVICE	This is a new service in support of Multipath.
$SET_SECURITY	The condition value, SS$_INVFILFOROP, has been added.
$SET_SYSTEM_EVENT	A new item code, SYSEVT$C_TDF_CHANGE, has been added.
$SHOW_INTRUSION	This service has been updated in support of Clusterwide Intrusion.
$WAKE	This service now accepts 64-bit addresses.

5.17 TCP/IP Files for SDA READ Command That Contain Global Symbols and Locations

Table 5–6 shows the TCP/IP files that contain global symbols for the VAX and Alpha SDA READ commands.

Table 5–6 Modules Containing Global Symbols and Data Structures Used by SDA

File	Contents
TCPIP$NET_GLOBALS.STB	Contains data structure definitions for TCP/IP Internet driver, execlet, and ACP data structures
TCPIP$NFS_GLOBALS.STB	Contains data structure definitions for TCP/IP NFS server
TCPIP$PROXY_GLOBALS.STB	Contains data structure definitions for TCP/IP proxy execlet
TCPIP$PWIP_GLOBALS.STB	Contains data structure definitions for TCP/IP PWIP driver, and ACP data structures
TCPIP$TN_GLOBALS.STB	Contains data structure definitions for TCP/IP TELNET/RLOGIN server driver data structures

These files are only available if TCP/IP services has been installed. They are found in SYS$SYSTEM, and are not automatically read in when you issue a READ/EXEC command.

Table 5–7 shows the TCP/IP files that define global locations within the Executive Image for the VAX SDA command.

Table 5–7 Modules Defining Global Locations Within the Executive Image

File	Contents
TCPIP$BGDRIVER.STB	TCP/IP Internet driver
TCPIP$INETACP.STB	TCP/IP Internet ACP
TCPIP$INTERNET_SERVICES.STB	TCP/IP Internet execlet
TCPIP$NFS_SERVICES.STB	Symbols for the TCP/IP NFS server
TCPIP$PROXY_SERVICES.STB	Symbols for the TCP/IP proxy execlet
TCPIP$PWIPACP.STB	TCP/IP PWIP ACP
TCPIP$PWIPDRIVER.STB	TCP/IP PWIP driver
TCPIP$TNDRIVER.STB	TCP/IP TELNET/RLOGIN server driver

These files are only available if TCP/IP services has been installed. They are found in SYS$SYSTEM, and are not automatically read in when you issue a READ/EXEC command.

For more detailed information, refer to the *OpenVMS VAX System Dump Analyzer Utility Manual* and the *OpenVMS Alpha System Analysis Tools Manual*.

5.18 Visual Threads Version 2.1 (Alpha)

Visual Threads is a unique tool that lets you debug and analyze multithreaded applications. You can use Visual Threads to automatically diagnose common problems associated with multithreading including deadlock, mutex, and thread usage errors. Also, you can use Visual Threads to monitor the thread-related performance of an application, helping you to identify bottlenecks or locking granularity problems. Visual Threads helps you identify problem areas in an application even if the application does not show specific symptoms.

Visual Threads includes the following features:

- Collects detailed information about significant thread-related state changes ("events").
- Analyzes common threading problems automatically based on predefined rules applied to the event stream.
- Rule customization for application-specific parameters and actions.
- Automatic statistics gathering, by sampling the event stream.
- Categories of analysis: data protection errors (race conditions), deadlocks, programming errors, lock activity, performance.
- Graphical visualization of the frequency of thread-related events and thread state, snapshots of historical program state, and object-specific graphs for each collected statistic.
- Lock activity profiling to reveal where various types of lock activity are occurring in your application, including: Number of Locks, Contended Locks, Locked Time, and Wait Time. Lock activity is collected and displayed for individual locks.
- Summarizes the program run and provides reports.
- Threads Snapshot view displays the historical state of threads represented at specific times in the main thread overview graph.

Programming Features
5.18 Visual Threads Version 2.1 (Alpha)

- Find and Filter support in the Event Window to allow you to quickly locate particular events.
- CPU Utilization Window shows the CPU percentage used by each thread.
- Thread Transitions Window depicts each state change for a detailed view.

For more information about these features, refer to the Visual Threads product documentation, which is available on the OpenVMS Alpha CD-ROM in directory [VISUAL_THREADS_021], or by using the online Help system.

6
Associated Products Features

This chapter describes significant new features of Compaq OpenVMS operating system associated products. For a listing and directory information on the OpenVMS associated products, refer to the *Guide to OpenVMS Version 7.3 CD-ROMs*.

6.1 Availability Manager

OpenVMS Version 7.3 contains Availability Manager Version 1.4. Soon after the release of OpenVMS Version 7.3, Availability Manager Version 2.0 will be announced on the following Availability Manager web site:

`http://www.openvms.compaq.com/openvms/products/availman/`

Version 2.0 will include the following new features:

- A new internal infrastructure supports new operating system features more easily and quickly.
- To support NUMA or OpenVMS ''RADs'' and to provide preliminary support for Wildfire/Galaxy, the following features have been implemented:
  - A new Memory view of OpenVMS Alpha V7.3 nodes displays RAD-related data.
  - When monitoring OpenVMS Alpha V7.3 nodes, Availability Manager displays a new single-process memory tab called ''RAD Counters.''
  - The CPU modes display includes the RAD for a CPU.
  - The CPU process list shows the home RAD for each process.
  - The Node summary display now includes the number of configured RADs, the system serial number, and the Galaxy ID of a node, if any.
- Displays now include additional switched LAN and NISCA data, when available.
- New user-defined event notifications have been implemented.
- A built-in browser now displays online help.
- A built-in Java runtime environment is now included. (In other words, you no longer need to install Java on the system.)
- ODS-5 file system support has been added.
- A new PGFLQUOTA process-level ''fix'' has been implemented.
- A simpler mechanism for site-specific configuration setup now exists.

6.2 Compaq Advanced Server V7.3 for OpenVMS (Alpha)

The Compaq Advanced Server Version 7.3 for OpenVMS is supported on Alpha systems only, and is the only version of the Advanced Server for OpenVMS supported on OpenVMS Alpha Version 7.3. New features include the following:

- Member server role (allowing the server to participate in Windows 2000 native-mode domains)
- Greater compatibility with a wide variety of clients and legacy applications, with support of:
  - Extended character sets, in addition to Extended File Specifications
  - Alias file names, created for shared files whose names do not comply with the more restricted file naming conventions of legacy applications such as MS-DOS
- Remote Windows NT printer management (SpoolSS) for printers shared on the Advanced Server for OpenVMS
- DNS for resolving NetBIOS names
- Cluster load balancing using DNS to resolve the server cluster alias name
- PCSI for installing the server
- Windows 2000 client and domain support

Earlier versions of the Advanced Server for OpenVMS (Versions 7.2 and 7.2A) must be upgraded to Version 7.3 to run on OpenVMS Alpha Version 7.3. Both the current and earlier versions of the Advanced Server for OpenVMS also run on OpenVMS Alpha Version 7.2-1.

For information about installing Advanced Server for OpenVMS, refer to the *Compaq Advanced Server for OpenVMS Server Installation and Configuration Guide* provided with the kit documentation.

To access Advanced Server V7.3 for OpenVMS on OpenVMS Alpha Version 7.3, clients must be licensed using the new Advanced Server V7.3 license PAK: PWLMXXXCA07.03. For more information, refer to the *Compaq Advanced Server for OpenVMS Guide to Managing Advanced Server Licenses*.

For information about the latest release of the PATHWORKS for OpenVMS (Advanced Server) product, supported on both OpenVMS Alpha and VAX Version 7.3 systems, see Section 6.9.

6.3 Compaq DECwindows Motif for OpenVMS

The *Compaq DECwindows Motif for OpenVMS* (DECwindows Motif), Version 1.2-6 kit for OpenVMS VAX and OpenVMS Alpha is now available. DECwindows Motif, Version 1.2-6 is a maintenance release that delivers a full range of changes and enhancements to your desktop. From faster batch scrolling to support for the Common Desktop Environment (CDE) screen saver and lock extensions, these changes are intended to provide you with a more efficient, flexible DECwindows Motif environment that is more in line with the OSF/Motif, MIT X11 Release 5 (X11 R5), and Common Desktop Environment (CDE) standards. For a full list of specific changes, enhancements, and corrections implemented in this release, refer to the *Compaq DECwindows Motif for OpenVMS Release Notes*.

6.4 Compaq DCE for OpenVMS

This section describes the enhancements in Compaq Distributed Computing Environment (DCE) for OpenVMS Version 7.3.

6.4.1 Compaq DCE Remote Procedure Call (RPC)

Beginning with OpenVMS Version 7.2-1, the NT Lan Manager security in DCE RPC is fully functional.

6.4.2 New Ethernet Device Support

If DCE RPC does not recognize the Ethernet device in your system, one new device may be added to the table of known devices by defining the system logical DCE$IEEE_802_DEVICE to be the device name of your Ethernet device.

For example, to define a single DE500 Ethernet device, set the logical as follows:

```
$ DEFINE/SYSTEM DCE$IEEE_802_DEVICE EWA0
```

6.4.3 For More DCE Information

Refer to the *OpenVMS Version 7.3 Release Notes* for important information about Compaq DCE for OpenVMS.

If you have the full DCE kit installed, you can use online help for additional information:

```
$ HELP DCE
$ HELP DCE$SETUP
$ HELP DCE_CDS
$ HELP DCE_DTS
$ HELP DCE_IDL
$ HELP DCE_RPC
$ HELP DCE_SECURITY
$ HELP DCE_THREADS
```

You can also refer to the following documentation:

- *Compaq DCE for OpenVMS VAX and OpenVMS Alpha Installation and Configuration Guide* (order number AA-PV4CE-TE)
- *Compaq DCE for OpenVMS VAX and OpenVMS Alpha Product Guide* (order number AA-PV4FE-TE)
- *Compaq DCE for OpenVMS VAX and OpenVMS Alpha Reference Guide* (order number AA-QHLZB-TE)

6.5 DECram for OpenVMS Version 3.0 (Alpha)

DECram Version 3.0 supports the OpenVMS for Alpha platform only. The following are the new features that can be found in this release:

- In DECram for OpenVMS Alpha Version 3.0, DECram's capability supports the use of shared memory for creation of RAM disks in an Adaptive Partitioned MultiProcessing (APMP) environment. This environment is also know as Compaq Galaxy Software Architecture.
- On OpenVMS Version 7.2-1H1 or higher, the limit on the DECram file size has been extended to 4,294,967,296 blocks.
- DECram for OpenVMS Version 3.0 is fully compatible with DECram Version 2.3. There can be any combination of these two versions of DECram in a VMScluster.

- Multiple DECram devices can be members of a Volume Shadowing for OpenVMS shadow set and can be served by Mass Storage Control Protocol (MSCP) or QIO served.
- Volume Shadowing for OpenVMS will support shadow sets composed of DECram devices and other disk class devices.
- A new DECram command interface (DECRAM>) can be used for creating, initializing, and mounting DECram disks.

DECram Version 3.0 and supporting documentation are included in the OpenVMS Version 7.3 CD-ROM in the [.DECRAM_030] directory.

6.6 Enterprise Capacity and Performance (ECP)

Beginning with OpenVMS Version 7.3, the following Enterprise Capacity and Performance (ECP) management tools will be provided at no additional costs. The ECP Data Collector for OpenVMS and ECP Performance Analyzer for OpenVMS will be available to customers who have a valid license to operate OpenVMS Version 6.2 or later. These products are available from the following World Wide Web site:

```
http://www.openvms.compaq.com/openvms/system_management.html
```

Software Support Services for these products are sold separately and are available on an incremental basis. Please contact your Compaq Services representative for further details.

6.6.1 ECP Collector for OpenVMS Version 5.4

ECP Collector for OpenVMS Version 5.4 gathers performance and capacity planning data on OpenVMS operating systems. OpenVMS data collection has three main criteria; the amount of performance data collected, the time interval, and the efficiency or amount of overhead impacting the system. ECP Collector for OpenVMS provides the following:

- Robust data collection set. It collects system metrics on over 250 OpenVMS performance parameters.
- Flexible data collection. The sampling rate of data can be tuned down to sub-second intervals.
- Low overhead. Audited production systems have routinely shown that ECP Collector for OpenVMS has less than a 1.5% impact on the CPU.

Satisfying the needs of Enterprise Management, ECP Collector for OpenVMS also contains an API that provides an interface for the access of performance data. This interface converts the contents of the .CPC data file generated by the data collector into a formatted, comma-separated ASCII file that can then be used for performance analysis and reporting programs.

6.6.2 ECP Performance Analyzer for OpenVMS Version 5.4

Compaq's ECP Analyzer for OpenVMS Version 5.4, which runs under Motif, analyzes the data provided by the ECP Collector for OpenVMS data collector. ECP Analyzer for OpenVMS provides the entry point into the data collector, and allows the user to select the sampling rate and to view the performance data in graphical format. The product provides historical information in standard graphs based upon the requested time interval. Graphs are provided for all common performance issues that need to be analyzed including the CPU, the memory, and

the I/O. ECP Analyzer for OpenVMS provides both graphic (MOTIF-based) and tabular reports for the data.

6.7 Kerberos for OpenVMS

Kerberos Version 1.0 for OpenVMS Alpha and OpenVMS VAX, based on MIT Kerberos Version 5 Release 1.0.5, is included on the OpenVMS Version 7.3 distribution media. (Kerberos documentation provided by MIT is included on the OpenVMS documentation CD-ROM in HTML format.)

Kerberos is a network authentication protocol designed to provide strong authentication for client/server applications by using secret-key cryptography.

Kerberos was created by the Massachusetts Institute of Technology as a solution to network security problems. The Kerberos protocol uses strong cryptography so that a client can prove its identity to a server (and vice versa) across an insecure network connection. After a client and server have used Kerberos to prove their identity, they can also encrypt all of their communications to assure privacy and data integrity.

General information about Kerberos is available from the following World Wide Web address:

`http://web.mit.edu/kerberos/www/`

6.7.1 New DCL Command KERBEROS

OpenVMS Kerberos is an authentication security product. It allows for user authentication for a wide range of communication programs such as RLOGIN, TELNET, and FTP.

Format:

```
KERBEROS [/ADMIN | /USER]
         [/INTERFACE=[DECWINDOWS | CHARACTER_CELL]]
```

Qualifiers:

`/ADMIN`

Activates the Kerberos administration utility for the selected interface.

`/USER (default)`

Activates the Kerberos user utility for the selected interface.

```
/INTERFACE=CHARACTER_CELL (default)
/INTERFACE=DECWINDOWS
```

Activates the display device requested, if available.

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

6.8 Universal LDAPv3 API (Alpha)

OpenVMS Version 7.3 includes the Lightweight Directory Access Protocol (LDAPv3) Application Programming Interface (API) that allows OpenVMS application developers, third-party applications, and users to access LDAP directories anywhere in the enterprise, intranet, extranet or Internet hosted by non-OpenVMS systems. The multi-threaded API will automatically support both 64-bit and 32-bit applications and be Common Object Model (COM) aware.

The universal LDAPv3 API is certified with Microsoft's Active Directory, Novell's NDS and Compaq's X.500 Version 4.0, and supports various security mechanisms including Kerberos V5 and Public Key Infrastructure (PKI).

The LDAPv3 kits are available from the following World Wide Web address:

`http://www.openvms.compaq.com/openvms/products/mgmt_agents/index.html`

For additional information on the LDAPv3 API, refer to the *OpenVMS Utility Routines Manual*.

6.9 Compaq PATHWORKS V6.0D for OpenVMS (Advanced Server)

Compaq PATHWORKS V6.0D for OpenVMS (Advanced Server) is the only PATHWORKS for OpenVMS server supported on OpenVMS Version 7.3 (in addition to Compaq Advanced Server V7.3 for OpenVMS). Earlier versions of PATHWORKS for OpenVMS servers must be upgraded. For more information, refer to the *OpenVMS Version 7.3 Release Notes.*

You can run PATHWORKS V6.0D for OpenVMS (Advanced Server) on either OpenVMS Alpha Versions 7.3, 7.2-1, or 6.2, or on OpenVMS VAX Versions 7.3, 7.2, or 6.2.

To access PATHWORKS V6.0D for OpenVMS (Advanced Server) on OpenVMS Version 7.3, clients must be licensed using the license PAK PWLMXXXCA06.00, PWLMXXXCA07.02, or PWLMXXXCA07.03. For more information, refer to the *Compaq Advanced Server for OpenVMS Guide to Managing Advanced Server Licenses*.

For information about the latest release of Compaq Advanced Server Version 7.3 for OpenVMS, see Section 6.2.

6.10 Compaq Service Tools and DECevent

Compaq Services new web-based service tool functionality is known as Web-Based Enterprise Services (WEBES). The Compaq System Tools CD-ROM included in the OpenVMS Version 7.3 CD-ROM package includes WEBES. (WEBES includes the Compaq Crash Analysis Tool (CCAT) and Compaq Analyze components.) This is the supported service tools for all AlphaServer DS, ES, and GS systems running OpenVMS, except for the AlphaServer GS60 and AlphaServer GS140 platforms. The AlphaServer GS60 and GS140 platforms must continue to use the DECevent diagnostic tool.

In addition to WEBES, the Compaq System Tools CD-ROM includes DECevent, DSNLINK, and the Revision and Configuration Management (RCM) tools.

DECevent and WEBES can be used together in a cluster.

Installation and documentation on the service tools are included on the Compaq System Tools CD-ROM. Use the following web site to access the most up-to-date service tool information:

`http://www.support.compaq.com/svctools/`

6.11 Compaq TCP/IP Services for OpenVMS Version 5.1

The Compaq TCP/IP Services for OpenVMS product is the Compaq implementation of the TCP/IP protocol suite and internet services for OpenVMS Alpha and OpenVMS VAX systems.

TCP/IP Services provides a comprehensive suite of functions and applications that support industry-standard protocols for heterogeneous network communications and resource sharing.

6.11.1 New Features and Changes

The new features of Compaq TCP/IP Services for OpenVMS Version 5.1 include:

- A new kernel, based on Compaq Tru64 UNIX Version 5.1.
- Support for Internet Protocol Version 6 (IPv6).
- DHCP client support.
- Xterminal support using XDM.
- Services that can be restarted individually.
- GATED enhancements.
- BIND dynamic updates management enhancements.
- Cluster failover for the BIND server.
- Cluster failover for the load broker.
- Updated SNMP that supports AgentX.
- SMTP enhancements, including:
  - AntiSPAM (configuration to control mail relay)
  - SMTP SFF (Send From File)
  - SMTP outbound alias
- Metric server logicals that can be changed without restarting the Metric server.
- The DHCP server can be configured to dynamically update the BIND database.
- TELNET client enhancements to support SNDLOC and NAWS.
- Support for the NFS V3 protocol in addition to the NFS V2 protocol in the NFS server.
- TCP options for improving certain performance characteristics.

For more information about configuring and managing these services, refer to the *Compaq TCP/IP Services for OpenVMS Management* guide provided with the TCP/IP Services for OpenVMS Version 5.1 software.

6.11.2 TCP/IP Services for OpenVMS Documentation

For installation instructions, refer to the *Compaq TCP/IP Services for OpenVMS Installation and Configuration* manual.

The TCP/IP Services for OpenVMS Release Notes provide version-specific information that supersedes the information in the documentation set. The features, restrictions, and corrections in this version of the software are described in the release notes. Always read the release notes before installing the software.

The TCP/IP Services for OpenVMS documentation set includes the following new items:

- *Compaq TCP/IP Services for OpenVMS Guide to IPv6*

  This manual describes the IPv6 environment, the roles of systems in this environment, the types and function of the different IPv6 addresses, and how to configure TCP/IP Services to access the 6bone network.

- *Compaq TCP/IP Services for OpenVMS Tuning and Troubleshooting*

  This manual provides information about how to isolate the causes of network problems and how to tune the TCP/IP Services software for the best performance.

- *Compaq TCP/IP Services for OpenVMS Management Command Quick Reference Card*

  This reference card summarizes the TCP/IP management commands, organizing them by function and component.

- *Compaq TCP/IP Services for OpenVMS UNIX Command Reference Card*

  This reference card describes how to use UNIX utilities on OpenVMS to manage TCP/IP services.

The following existing TCP/IP Services for OpenVMS manuals have been updated for V5.1:

- *Compaq TCP/IP Services for OpenVMS Installation and Configuration*
- *Compaq TCP/IP Services for OpenVMS Management*
- *Compaq TCP/IP Services for OpenVMS Management Command Reference*
- *Compaq TCP/IP Services for OpenVMS Sockets API and System Services Programming*
- *Compaq TCP/IP Services for OpenVMS SNMP Programming and Reference*

Part II

Overview of OpenVMS Documentation

7

OpenVMS Documentation Overview

This chapter outlines the changes to the OpenVMS documentation from the previous version.

7.1 OpenVMS Documentation Changes

Table 7–1 lists the changes to the OpenVMS documentation for OpenVMS Version 7.3.

Table 7–1 Changes to Documentation for OpenVMS Version 7.3

Area of Change	Description
	Expanded Online Offerings
Documentation CD-ROM	The OpenVMS documentation is now available on a single ISO9660 Level 2 CD-ROM. This CD-ROM can be used on OpenVMS, Windows, and Macintosh systems. For more information about the documentation CD-ROM, see Section 8.2.
Online formats	OpenVMS documentation is now available in PDF as well as HTML formats on the documentation CD-ROM.
Adobe Acrobat Reader	The documentation CD-ROM includes two versions of the Adobe Acrobat Reader. One is an executable file for PCs; one is a ZIP file that can be installed on OpenVMS Alpha systems running Java Version 1.1.8-5.
Online help	Reference information for the following OpenVMS utility routines is now included in online help under `RTL_Routines`: Access Control List (ACL) Editor (ACLEDIT$) Backup (BACKUP) (BACKUP$) Command Language (CLI) (CLI$) Command File Qualifier (UTIL$) Convert (CONVERT) (CONV$) Data Compression/Expansion (DCX) (DCX$) DEC Text Processing Utility (DECTPU) (TPU$) EDT (EDT$EDIT; FILEIO; WORKIO; XLATE) File Definition Language (FDL) (FDL$) Librarian (LBR) (LBR$) LOGINOUT (LGI) (LGI$) Mail Utility (MAIL) (MAIL$) National Character Set (NCS) (NCS$) Print Symbiont Modification (PSM) (PSM$; USER-*x*) Symbiont/Job Controller Interface (SMB) (SMB$) Sort/Merge (SOR) (SOR$)

(continued on next page)

Table 7–1 (Cont.) Changes to Documentation for OpenVMS Version 7.3

Area of Change	Description
	Expanded Online Offerings
Master Index	The *OpenVMS Master Index* manual in ASCII text format is now included on the documentation CD-ROM.
	New Manual
Availability Manager User's Guide	This new manual for Version 7.3 describes how to use the Compaq Availability Manager system management tool, from either an OpenVMS Alpha or a Windows node, to monitor one or more OpenVMS nodes on an extended local area network (LAN) or to target a specific node or process for detailed analysis.
	Revised Titles
Title changes	Note the following titles changes for Version 7.3: • *Compaq C Run-Time Library Reference Manual for OpenVMS Systems* Revised to reflect the Compaq C product name. • *Compaq Portable Mathematics Library* Revised to reflect the Compaq name. • *Guide to POSIX Threads Library* Formerly entitled *Guide to DECthreads*. • *OpenVMS Alpha Partitioning and Galaxy Guide* Formerly entitled *OpenVMS Galaxy Guide*. • *OpenVMS MACRO-32 Porting and User's Guide* Formerly entitled *Porting VAX MACRO Code to OpenVMS Alpha*. • *OpenVMS Version 7.3 New Features and Documentation Overview* Now includes information about the OpenVMS documentation offerings.

(continued on next page)

Table 7–1 (Cont.) Changes to Documentation for OpenVMS Version 7.3

Area of Change	Description
	Restructured Documentation
OpenVMS Programming Concepts Manual	The *OpenVMS Programming Concepts Manual* has been restructured and expanded. It now consists of the following two volumes: • *OpenVMS Programming Concepts Manual, Volume I* Contains an introduction to OpenVMS programming followed by three parts: Process and Synchronization; Interrupts and Condition Handling; and Addressing and Memory Management. • *OpenVMS Programming Concepts Manual, Volume II* Contains two additional parts: Calling a System Routine; and I/O, System, and Programming Routines.
DCL command MOUNT	Reference information about the MOUNT command is now located in the *OpenVMS DCL Dictionary: A–M*.
Information about OpenVMS documentation	The *Overview of OpenVMS Documentation* manual no longer exists as a separate manual. Instead, information about OpenVMS documentation is included in the *OpenVMS Version 7.3 New Features and Documentation Overview*.
	Archived Manuals
Four archived manuals	Beginning with Version 7.3, the following manuals are archived. The online versions are included with other archived manuals in a separate directory on the OpenVMS documentation CD-ROM. • *Migrating an Application from OpenVMS VAX to OpenVMS Alpha* • *OpenVMS Alpha Guide to 64-Bit Addressing and VLM Features* Portions are now included in the *OpenVMS Programming Concepts Manual, Volume I*. • *OpenVMS Programming Interfaces: Calling a System Routine* Portions are now included in the *OpenVMS Programming Concepts Manual, Volume II*. • *TCP/IP Networking on OpenVMS Systems* For information about using TCP/IP on OpenVMS systems, see the *OpenVMS System Manager's Manual* and the documentation set for the Compaq TCP/IP Services for OpenVMS product.
Addition to the archived manuals offering	The *Standard TECO Text Editor and Corrector for the VAX, PDP-11, PDP-10, and PDP-8* manual is now included with the archived manuals on the documentation CD-ROM. Note that this manual is not available in printed form.

8

OpenVMS Printed and Online Documentation

This chapter outlines the following OpenVMS documentation components:

- Printed documentation (Section 8.1)
- Online documentation on CD-ROM (Section 8.2)
- Online documentation on the OpenVMS web site (Section 8.3)
- Online Help (Section 8.4)

8.1 Printed Documentation

This section describes the OpenVMS printed documentation offerings:

- Upgrade and installation documentation that is included with your OpenVMS software media kit (Section 8.1.1)
- The OpenVMS Full and Base Documentation sets (Section 8.1.2)
- Documentation for System Integrated Products (SIPs) (Section 8.1.3)
- Archived manuals (Section 8.1.4)

8.1.1 OpenVMS Media Kit Documentation

Upgrade and installation manuals and the *OpenVMS License Management Utility Manual* are included in your OpenVMS media kit.

OpenVMS Alpha customers receive the *OpenVMS Alpha Version 7.3 Upgrade and Installation Manual*. This manual contains step-by-step upgrade and installation information as well as device-naming and booting information.

OpenVMS VAX customers receive the *OpenVMS VAX Version 7.3 Upgrade and Installation Manual*, the primary source for step-by-step upgrade and installation procedures using the VMSINSTAL utility.

Table 8–1 lists the installation manuals included in the OpenVMS media kit. You can also order the manuals individually. Note that the manuals in the media kit are not part of the OpenVMS Full Documentation set.

Table 8–1 OpenVMS Media Kit Manuals

Manual	Order Number
Upgrade and Installation Manuals	
OpenVMS Alpha Version 7.3 Upgrade and Installation Manual	AA-QSE8D-TE
OpenVMS VAX Version 7.3 Upgrade and Installation Manual	AA-QSBQD-TE

(continued on next page)

Table 8–1 (Cont.) OpenVMS Media Kit Manuals

Manual	Order Number
Upgrade and Installation Manuals	
OpenVMS License Management Utility Manual	AA-PVXUF-TK
Guide to OpenVMS Version 7.3 CD–ROMs	AA-QSBRE-TE

8.1.2 OpenVMS Documentation Sets

OpenVMS documentation is available in the following documentation sets:

Documentation Set	Description	Order Number
Full set	Intended for users who need extensive explanatory information for all major OpenVMS resources. Contains all the OpenVMS documentation in one offering. Includes the Base Documentation set.	QA-001AA-GZ.7.3
Base set	Subset of the Full Documentation set. Intended for general users and system managers of small standalone systems. Includes the most commonly used OpenVMS manuals.	QA-09SAA-GZ.7.3

In addition to ordering the Full or Base Documentation set, you can order any OpenVMS manual individually.

Table 8–2 lists the manuals in the OpenVMS Base and Full Documentation sets. For a description of each manual, see Section 9.2.

Table 8–2 OpenVMS Full Documentation Set (QA-001AA-GZ.7.3)

Manual	Order Number
OpenVMS Base Documentation Set	**QA-09SAA-GZ.7.3**
OpenVMS DCL Dictionary: A–M	AA-PV5KG-TK
OpenVMS DCL Dictionary: N–Z	AA-PV5LG-TK
OpenVMS Guide to System Security	AA-Q2HLE-TE
OpenVMS Master Index	AA-QSBSD-TE
OpenVMS System Management Utilities Reference Manual: A–L	AA-PV5PF-TK
OpenVMS System Management Utilities Reference Manual: M–Z	AA-PV5QF-TK
OpenVMS System Manager's Manual, Volume 1: Essentials	AA-PV5MF-TK
OpenVMS System Manager's Manual, Volume 2: Tuning, Monitoring, and Complex Systems	AA-PV5NF-TK
OpenVMS User's Manual	AA-PV5JE-TK
OpenVMS Version 7.3 New Features and Documentation Overview	AA-QSBFD-TE
OpenVMS Version 7.3 Release Notes	AA-QSBTD-TE

(continued on next page)

Table 8–2 (Cont.) OpenVMS Full Documentation Set (QA-001AA-GZ.7.3)

Manual	Order Number
Continuation of Full Documentation Set	**QA-001AA-GZ.7.3**
Availability Manager User's Guide	AA-RNSJA-TE
Compaq C Run-Time Library Utilities Reference Manual	AA-R238C-TE
Compaq Portable Mathematics Library	AA-PV6VE-TE
DECamds User's Guide	AA-Q3JSE-TE
DEC Text Processing Utility Reference Manual	AA-PWCCD-TE
Extensible Versatile Editor Reference Manual	AA-PWCDD-TE
Guidelines for OpenVMS Cluster Configurations	AA-Q28LE-TK
Guide to Creating OpenVMS Modular Procedures	AA-PV6AD-TK
Guide to OpenVMS File Applications	AA-PV6PD-TK
Guide to POSIX Threads Library	AA-QSBPD-TE
Guide to the DEC Text Processing Utility	AA-PWCBD-TE
OpenVMS Alpha Partitioning and Galaxy Guide	AA-REZQC-TE
OpenVMS Alpha Guide to Upgrading Privileged-Code Applications	AA-QSBGD-TE
OpenVMS Alpha System Analysis Tools Manual	AA-REZTB-TE
OpenVMS Calling Standard	AA-QSBBD-TE
OpenVMS Cluster Systems	AA-PV5WE-TK
OpenVMS Command Definition, Librarian, and Message Utilities Manual	AA-QSBDD-TE
OpenVMS Debugger Manual	AA-QSBJD-TE
OpenVMS Delta/XDelta Debugger Manual	AA-PWCAD-TE
OpenVMS Guide to Extended File Specifications	AA-REZRB-TE
OpenVMS I/O User's Reference Manual	AA-PV6SD-TK
OpenVMS Linker Utility Manual	AA-PV6CD-TK
OpenVMS MACRO-32 Porting and User's Guide	AA-PV64D-TE
OpenVMS Management Station Overview and Release Notes	AA-QJGCD-TE
OpenVMS Performance Management	AA-R237C-TE
OpenVMS Programming Concepts Manual, Volume I	AA-RNSHA-TK
OpenVMS Programming Concepts Manual, Volume II	AA-PV67E-TK
OpenVMS Record Management Services Reference Manual	AA-PV6RD-TK
OpenVMS Record Management Utilities Reference Manual	AA-PV6QD-TK
OpenVMS RTL General Purpose (OTS$) Manual	AA-PV6HD-TK
OpenVMS RTL Library (LIB$) Manual	AA-QSBHD-TE
OpenVMS RTL Screen Management (SMG$) Manual	AA-PV6LD-TK
OpenVMS RTL String Manipulation (STR$) Manual	AA-PV6MD-TK
OpenVMS System Messages: Companion Guide for Help Message Users	AA-PV5TD-TK
OpenVMS System Services Reference Manual: A–GETUAI	AA-QSBMD-TE
OpenVMS System Services Reference Manual: GETUTC–Z	AA-QSBND-TE
OpenVMS Utility Routines Manual	AA-PV6EE-TK

(continued on next page)

Table 8–2 (Cont.) OpenVMS Full Documentation Set (QA-001AA-GZ.7.3)

Manual	Order Number
Continuation of Full Documentation Set	**QA-001AA-GZ.7.3**
OpenVMS VAX RTL Mathematics (MTH$) Manual	AA-PVXJD-TE
OpenVMS VAX System Dump Analyzer Utility Manual	AA-PV6TD-TE
POLYCENTER Software Installation Utility Developer's Guide	AA-Q28MD-TK
VAX MACRO and Instruction Set Reference Manual	AA-PS6GD-TE
Volume Shadowing for OpenVMS	AA-PVXMG-TE

8.1.3 Documentation for System Integrated Products

The following System Integrated Products (SIPs) are included in the OpenVMS software. You must purchase separate licenses to enable them.

- Compaq Galaxy Software Architecture on OpenVMS Alpha
- OpenVMS Clusters
- RMS Journaling for OpenVMS
- Volume Shadowing for OpenVMS

The OpenVMS Cluster and Volume Shadowing documentation is included in the OpenVMS Full Documentation set.

You must purchase RMS Journaling documentation separately. Use the following part number to order the *RMS Journaling for OpenVMS Manual*: AA-JG41C-TE.

8.1.4 Archived OpenVMS Documentation

OpenVMS continuously updates, revises, and enhances the OpenVMS operating system documentation. From time to time, manuals are archived. You can access the archived manuals online from the documentation CD-ROM or from the following web site:

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

For a list of the archived OpenVMS manuals, see Section 9.5.

8.2 Online Documentation on CD-ROM

Beginning with OpenVMS Version 7.3, online documentation for the OpenVMS operating system and many associated products is provided on a single ISO9660 Level 2 CD-ROM. This CD-ROM is readable on OpenVMS, Windows, and Macintosh systems.

8.2.1 Online Formats

The documentation CD-ROM contains documentation in a number of formats, including HTML, PDF, PostScript, and ASCII text. Bookreader files are no longer available on the documentation CD-ROM.

8.2.2 PDF Viewers

Tools are supplied on the documentation CD-ROM for viewing PDF files on OpenVMS Alpha systems and Windows systems.

For OpenVMS Alpha users, the Adobe Acrobat Viewer (for Java) is provided. Instructions for installing this Java implementation of the Acrobat Viewer are provided in a README file on the CD-ROM.

For users of Windows systems, an executable file for the Adobe Acrobat Viewer is provided. This self-extracting file can be installed on a personal computer running Windows.

For information about how to access documents on the documentation CD-ROM and about the PDF viewers, refer to the *Guide to OpenVMS Version 7.3 CD–ROMs*.

8.3 Online Documentation on the OpenVMS Web Site

You can access OpenVMS manuals in various online formats from the following OpenVMS web site:

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

The OpenVMS web site contains links to current versions of manuals in the OpenVMS Full Documentation set as well as manuals for selected layered products.

8.4 Online Help

The OpenVMS operating system provides online help for the commands, utilities, and system routines documented in the Full Documentation set.

You can use the Help Message facility to quickly access online descriptions of system messages. In addition, you can add your own source files, such as messages documentation that you have written to the Help Message database. The *OpenVMS System Messages: Companion Guide for Help Message Users* manual explains how to use the Help Message facility. You can also access DCL Help for Help Message by entering:

```
$ HELP HELP/MESSAGE
```

9
Descriptions of OpenVMS Manuals

This chapter provides summary descriptions for the following OpenVMS documentation:

- Manuals in the OpenVMS Media Kit (Section 9.1)
- Manuals in the OpenVMS Full Documentation set (Section 9.2)
- Manual about how to write device drivers for OpenVMS Alpha systems (Section 9.3)
- RMS Journaling manual (Section 9.4)
- Archived manuals (Section 9.5)

9.1 Manuals Included in the OpenVMS Media Kit

Guide to OpenVMS Version 7.3 CD–ROMs

Provides information about the OpenVMS Alpha and OpenVMS VAX operating system and documentation CD-ROMs. Lists the contents of the OpenVMS Version 7.3 CD-ROM package, includes pointers to installation information, and gives instructions about how to access manuals on the Documentation CD-ROM.

OpenVMS License Management Utility Manual

Describes the License Management Facility (LMF), the OpenVMS license management tool. LMF includes the License Management Utility (LICENSE) and the command procedure VMSLICENSE.COM, which is used to register, manage, and track software licenses on line.

OpenVMS Alpha Version 7.3 Upgrade and Installation Manual

Provides step-by-step instructions for installing and upgrading the OpenVMS Alpha operating system on Alpha computers. Includes information about booting, shutdown, backup, and licensing procedures.

OpenVMS VAX Version 7.3 Upgrade and Installation Manual

Provides step-by-step instructions for installing and upgrading the OpenVMS VAX operating system on OpenVMS VAX computers. Includes information about booting, shutdown, backup, and licensing procedures.

9.2 Manuals in the OpenVMS Documentation Set

9.2.1 OpenVMS Base Documentation Set

OpenVMS DCL Dictionary

Describes the DIGITAL Command Language (DCL) and provides an alphabetical listing of detailed reference information and examples for all DCL commands and lexical functions. This manual is in two volumes.

OpenVMS Guide to System Security

Describes the security features available in the OpenVMS Alpha and VAX operating systems. Explains the purpose and proper application of each feature in the context of specific security needs.

OpenVMS Master Index

Offers an edited compilation of indexes from the manuals in the OpenVMS Full Documentation set.

OpenVMS System Management Utilities Reference Manual

Presents reference information about the utilities you can use to perform system management tasks on your system as well as the tools to control and monitor system access and resources. Includes a description of the AUTOGEN command procedure. This manual is in two volumes.

OpenVMS System Manager's Manual, Volume 1: Essentials

Provides instructions for setting up and maintaining routine operations such as starting up the system, installing software, and setting up print and batch queues. Also explains routine disk and magnetic tape operations.

OpenVMS System Manager's Manual, Volume 2: Tuning, Monitoring, and Complex Systems

Describes how to configure and control the network, how to monitor the system, and how to manage system parameters. Also includes information about OpenVMS Cluster systems, network environments, and DECdtm functionality.

OpenVMS User's Manual

Provides an overview of the operating system and presents basic concepts, task information, and reference information that allow you to perform daily computing tasks. Describes how to work with files and directories. Also includes these additional topics:

- Sending messages with the Mail utility and the Phone utility
- Using the Sort/Merge utility
- Using logical names and symbols
- Writing command procedures
- Editing files with the EVE and EDT text editors

OpenVMS Version 7.3 New Features and Documentation Overview

Describes new and improved components for the Alpha and VAX operating systems for the Version 7.3 release. Includes information about OpenVMS documentation changes for Version 7.3 as well as the printed and online OpenVMS documentation offerings.

OpenVMS Version 7.3 Release Notes

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

9.2.2 Continuation of Full Documentation Set

Availability Manager User's Guide

Describes how to use the Compaq Availability Manager system management tool, from either an OpenVMS Alpha or a Windows node, to monitor one or more OpenVMS nodes on an extended local area network (LAN) or to target a specific node or process for detailed analysis.

Compaq C Run-Time Library Utilities Reference Manual

Provides detailed usage and reference information about the Compaq C Run-Time Library utilities for managing localization and time zone data in international software applications.

Compaq Portable Mathematics Library

Documents the mathematics routines in the Compaq Portable Mathematics Library (DPML), supplied only with OpenVMS Alpha systems. VAX programmers should refer to the *OpenVMS VAX RTL Mathematics (MTH$) Manual*.

DECamds User's Guide

Provides information for installing and using the DECamds software. DECamds is a system management tool that lets you monitor, diagnose, and track events in OpenVMS system and OpenVMS Cluster environments.

DEC Text Processing Utility Reference Manual

Describes the DEC Text Processing Utility (DECTPU) and provides reference information about the EDT Keypad Emulator interfaces to DECTPU.

Extensible Versatile Editor Reference Manual

Contains command reference information about the EVE text editor. Also provides a cross-reference between EDT and EVE commands.

Guidelines for OpenVMS Cluster Configurations

This manual provides information to help you choose systems, interconnects, storage devices, and software. It can help you configure these components to achieve high availability, scalability, performance, and ease of system management. Detailed directions using SCSI and Fibre Channel in an OpenVMS Cluster system are also included in this manual.

Guide to Creating OpenVMS Modular Procedures

Describes how to perform a complex programming task by dividing it into modules and coding each module as a separate procedure.

Guide to OpenVMS File Applications

Contains guidelines for designing, creating, and maintaining efficient data files by using Record Management Services (RMS). This manual is intended for application programmers and designers responsible for programs that use RMS files, especially if performance is an important consideration.

Guide to POSIX Threads Library

Describes the POSIX Threads Library (formerly named DECthreads) package, Compaq's multithreading run-time libraries. Use the routines in this package to create and control multiple threads of execution within the address space provided by a single process. Offering both usage tips and reference synopses, this document describes three interfaces: routines that conform to the IEEE POSIX 1003.1c standard (called *pthread*), routines that provide thread-related services in nonthreaded applications (called thread-independent services or *tis*), and a set of Compaq proprietary routines (called *cma*) that provide a stable, upwardly compatible interface.

Guide to the DEC Text Processing Utility

Provides an introduction to developing DECTPU programs.

OpenVMS Alpha Partitioning and Galaxy Guide

Provides complete details about how to use all of the OpenVMS Galaxy features and capabilities available in OpenVMS Alpha Version 7.3. Includes procedures for creating, managing, and using OpenVMS Galaxy computing environments on AlphaServer 8400, 8200, and 4100 systems.

OpenVMS Alpha Guide to Upgrading Privileged-Code Applications

Explains the OpenVMS Alpha Version 7.0 changes that might impact Alpha privileged-code applications and device drivers as a result of the OpenVMS Alpha 64-bit virtual addressing and kernel threads support provided in OpenVMS Alpha Version 7.0.

Privileged-code applications from versions prior to OpenVMS Alpha Version 7.0 might require the source-code changes described in this guide.

OpenVMS Alpha System Analysis Tools Manual

Describes the following system analysis tools in detail, while also providing a summary of the dump off system disk (DOSD) capability and the DELTA/XDELTA debugger:

- System Dump Analysis (SDA)
- System code debugger (SCD)
- System dump debugger (SDD)
- Watchpoint utility

Intended primarily for the system programmer who must investigate the causes of system failures and debug kernel mode code, such as a device driver.

OpenVMS Calling Standard

Documents the calling standard for the OpenVMS Alpha and VAX operating system.

OpenVMS Cluster Systems

Describes procedures and guidelines for configuring and managing OpenVMS Cluster systems. Also describes how to provide high availability, building-block growth, and unified system management across clustered systems.

OpenVMS Command Definition, Librarian, and Message Utilities Manual

Contains descriptive and reference information about the following utilities:

- Command Definition utility
- Librarian utility
- Message utility

OpenVMS Debugger Manual

Explains the features of the OpenVMS Debugger for programmers.

OpenVMS Delta/XDelta Debugger Manual

Describes the Delta/XDelta utility used to debug programs that run in privileged processor mode or at an elevated interrupt priority level.

OpenVMS Guide to Extended File Specifications

Provides an overview of Extended File Specifications and describes the overall differences and impact Extended File Specifications introduce to the OpenVMS environment.

OpenVMS I/O User's Reference Manual

Contains the information that system programmers need to program I/O operations using the device drivers that are supplied with the operating system.

OpenVMS Linker Utility Manual

Describes how to use the Linker utility to create images that run on OpenVMS systems. Also explains how to control a link operation with link qualifiers and link options.

OpenVMS MACRO-32 Porting and User's Guide

Describes how to port existing VAX MACRO assembly language code to an OpenVMS Alpha system by using the features of the MACRO-32 compiler. Also documents how to use the compiler's 64-bit addressing support.

OpenVMS Management Station Overview and Release Notes

Provides an overview of OpenVMS Management Station and describes how to get started using the software. OpenVMS Management Station is a powerful, Microsoft Windows based management tool for system managers and others who perform user account and printer management tasks on OpenVMS systems. OpenVMS Management Station provides a comprehensive user interface to OpenVMS user account and printer management. Also includes release notes for OpenVMS Management Station.

OpenVMS Performance Management

Introduces and explains the techniques used to optimize performance on an OpenVMS system.

OpenVMS Programming Concepts Manual

Describes concepts such as process creation, kernel threads and the kernel threads process structure, interprocess communication, process control, data sharing, condition handling, and ASTs. This two-volume manual uses system services, utility routines, and run-time library (RTL) routines to illustrate mechanisms for utilizing OpenVMS features.

OpenVMS Record Management Services Reference Manual

Provides reference and usage information for all programmers who use RMS data files.

OpenVMS Record Management Utilities Reference Manual

Contains descriptive and reference information about the following RMS utilities:

- Analyze/RMS_File utility
- Convert and Convert/Reclaim utilities
- File Definition Language facility

OpenVMS RTL General Purpose (OTS$) Manual

Documents the general-purpose routines contained in the OTS$ facility of the OpenVMS Run-Time Library. Indicates which routines are specific to Alpha or VAX, as well as how routines function differently on each system.

OpenVMS RTL Library (LIB$) Manual

Documents the general-purpose routines contained in the LIB$ facility of the OpenVMS Run-Time Library. Indicates which routines are specific to Alpha or VAX, as well as how routines function differently on each system.

OpenVMS RTL Screen Management (SMG$) Manual

Documents the screen management routines contained in the SMG$ facility of the OpenVMS Run-Time Library. Indicates which routines are specific to Alpha or VAX, as well as how routines function differently on each system.

OpenVMS RTL String Manipulation (STR$) Manual

Documents the string manipulation routines contained in the STR$ facility of the OpenVMS Run-Time Library. Indicates which routines are specific to Alpha or VAX, as well as how routines function differently on each system.

OpenVMS System Messages: Companion Guide for Help Message Users

Describes features of the Help Message facility, a tool that you can use to display message descriptions. Describes the HELP/MESSAGE command and qualifiers and also includes detailed information about customizing the Help Message database. Also provides descriptions of messages that can occur when the system and Help Message are not fully operable.

OpenVMS System Services Reference Manual

Presents the set of routines that the operating system uses to control resources, allow process communication, control I/O, and perform other such operating system functions. This manual is in two volumes.

OpenVMS Utility Routines Manual

Describes the routines that allow a program to use the callable interface of selected OpenVMS utilities.

OpenVMS VAX RTL Mathematics (MTH$) Manual

Documents the mathematics routines contained in the MTH$ facility of the OpenVMS Run-Time Library, which is relevant only to programmers using OpenVMS VAX. (Alpha programmers should refer to *Compaq Portable Mathematics Library*.)

OpenVMS VAX System Dump Analyzer Utility Manual

Explains how to use the System Dump Analyzer utility to investigate system failures and examine a running OpenVMS VAX system. VAX programmers should refer to this manual; Alpha programmers should refer to the *OpenVMS Alpha System Dump Analyzer Utility Manual*.

POLYCENTER Software Installation Utility Developer's Guide

Describes the procedure and provides guidelines for developing software products that will be installed using the POLYCENTER Software Installation utility. Intended for developers who are designing installation procedures for software products layered on the OpenVMS operating system.

VAX MACRO and Instruction Set Reference Manual

Documents both the assembler directives of VAX MACRO and the VAX instruction set.

Volume Shadowing for OpenVMS

Describes how to provide high data availability with phase II volume shadowing.

9.3 OpenVMS Alpha Device Driver Manual

Writing OpenVMS Alpha Device Drivers in C

Provides definitive information about writing device drivers in the C programming language for devices connected to Alpha processors. *Writing OpenVMS Alpha Device Drivers in C* is available from Digital Press. For more information, access the following web site:

`http://www.bh.com/digitalpress`

9.4 RMS Journaling Manual

RMS Journaling for OpenVMS Manual

Describes the three types of RMS Journaling as well as other OpenVMS components that support RMS Journaling. This manual also describes the RMS Recovery utility (which is used to recover data saved using journaling), the transaction processing system services, and system management tasks required when using RMS Journaling.

9.5 Archived Manuals

Table 9–1 lists the OpenVMS manuals that have been archived. Please note the following:

- Most archived manuals can be ordered separately in printed format from Compaq. To order archived manuals, call 800-344-4825 and have the book title and part number.
- Archived manuals are available in various formats (PostScript and PDF) on the documentation CD-ROM.
- Most information from the archived manuals has been incorporated in other documents or online help.

Table 9–1 Archived OpenVMS Manuals

Manual	Order Number
A Comparison of System Management on OpenVMS AXP and OpenVMS VAX	AA-PV71B-TE
Building Dependable Systems: The OpenVMS Approach	AA-PV5YB-TE
Creating an OpenVMS Alpha Device Driver from an OpenVMS VAX Device Driver	AA-R0Y8A-TE
Creating an OpenVMS AXP Step 2 Device Driver from a Step 1 Device Driver	AA-Q28TA-TE
Creating an OpenVMS AXP Step 2 Device Driver from an OpenVMS VAX Device Driver	AA-Q28UA-TE
Guide to OpenVMS AXP Performance Management	AA-Q28WA-TE
Guide to OpenVMS Performance Management	AA-PV5XA-TE
Migrating an Application from OpenVMS VAX to OpenVMS Alpha	AA-KSBKB-TE
Migrating an Environment from OpenVMS VAX to OpenVMS Alpha	AA-QSBLA-TE
Migrating to an OpenVMS AXP System: Planning for Migration	AA-PV62A-TE

(continued on next page)

Table 9–1 (Cont.) Archived OpenVMS Manuals

Manual	Order Number
Migrating to an OpenVMS AXP System: Recompiling and Relinking Applications	AA-PV63A-TE
OpenVMS Alpha Guide to 64-Bit Addressing and VLM Features	AA-QSBCC-TE
OpenVMS Alpha System Dump Analyzer Utility Manual	AA-PV6UC-TE
OpenVMS AXP Device Support: Developer's Guide	AA-Q28SA-TE
OpenVMS AXP Device Support: Reference	AA-Q28PA-TE
OpenVMS Bad Block Locator Utility Manual	AA-PS69A-TE
OpenVMS Compatibility Between VAX and Alpha	AA-PYQ4C-TE
OpenVMS Developer's Guide to VMSINSTAL	AA-PWBXA-TE
OpenVMS DIGITAL Standard Runoff Reference Manual	AA-PS6HA-TE
OpenVMS EDT Reference Manual	AA-PS6KA-TE
OpenVMS Exchange Utility Manual	AA-PS6AA-TE
OpenVMS Glossary	AA-PV5UA-TK
OpenVMS National Character Set Utility Manual	AA-PS6FA-TE
OpenVMS Obsolete Features Manual	AA-PS6JA-TE
OpenVMS Programming Environment Manual	AA-PV66B-TK
OpenVMS Programming Interfaces: Calling a System Routine	AA-PV68B-TK
OpenVMS RTL DECtalk (DTK$) Manual	AA-PS6CA-TE
OpenVMS RTL Parallel Processing (PPL$) Manual	AA-PV6JA-TK
OpenVMS Software Overview	AA-PVXHB-TE
OpenVMS SUMSLP Utility Manual	AA-PS6EA-TE
OpenVMS System Messages and Recovery Procedures Reference Manual: A–L	AA-PVXKA-TE
OpenVMS System Messages and Recovery Procedures Reference Manual: M–Z	AA-PVXLA-TE
OpenVMS Terminal Fallback Utility Manual	AA-PS6BA-TE
OpenVMS VAX Card Reader, Line Printer, and LPA11–K I/O User's Reference Manual	AA-PVXGA-TE
OpenVMS VAX Device Support Manual	AA-PWC8A-TE
OpenVMS VAX Device Support Reference Manual	AA-PWC9A-TE
OpenVMS VAX Patch Utility Manual	AA-PS6DA-TE
OpenVMS Wide Area Network I/O User's Reference Manual	AA-PWC7A-TE
PDP-11 TECO User's Guide	AA-K420B-TC
POLYCENTER Software Installation Utility User's Guide	AA-Q28NA-TK
TCP/IP Networking on OpenVMS Systems	AA-QJGDB-TE
Standard TECO Text Editor and Corrector for the VAX, PDP-11, PDP-10, and PDP-8	Available only on CD-ROM

Table 9–2 lists the networking manuals and installation supplements that have been archived.

Table 9–2 Archived Networking Manuals and Installation Supplements

Manual	Order Number
DECnet for OpenVMS Guide to Networking	AA-PV5ZA-TK
DECnet for OpenVMS Network Management Utilities	AA-PV61A-TK
DECnet for OpenVMS Networking Manual	AA-PV60A-TK
OpenVMS VAX Upgrade and Installation Supplement: VAX 8820, 8830, 8840	AA-PS6MA-TE
OpenVMS VAX Upgrade and Installation Supplement: VAX 8200, 8250, 8300, 8350	AA-PS6PA-TE
OpenVMS VAX Upgrade and Installation Supplement: VAX 8530, 8550, 8810 (8700), and 8820–N (8800)	AA-PS6QA-TE
OpenVMS VAX Upgrade and Installation Supplement: VAX 8600, 8650	AA-PS6UA-TE
VMS Upgrade and Installation Supplement: VAX-11/780, 785	AA-LB29B-TE
VMS Upgrade and Installation Supplement: VAX-11/750	AA-LB30B-TE

Descriptions of the archived OpenVMS manuals are as follows:

A Comparison of System Management on OpenVMS AXP and OpenVMS VAX

Discusses system management tools, the impact of Alpha page sizes on system management operations, the system directory structure, interoperability issues, and performance information. Designed for system managers who need to learn quickly how to manage an OpenVMS Alpha system.

Building Dependable Systems: The OpenVMS Approach

Offers practical information about analyzing the dependability requirements of your business applications and deciding how to use your computing systems to support your dependability goals. This information is complemented by technical summaries of the dependability features of OpenVMS and related hardware and layered software products.

Creating an OpenVMS Alpha Device Driver from an OpenVMS VAX Device Driver

Describes the procedures for converting a device driver used on OpenVMS VAX to a device driver that runs on OpenVMS Alpha. This book also contains data structures, routines, and macros for maintaining an Alpha driver written in Macro-32.

Creating an OpenVMS AXP Step 2 Device Driver from a Step 1 Device Driver

Provides information for upgrading a Step 1 device driver (used in earlier versions of OpenVMS AXP) to a Step 2 device driver. A Step 2 device driver is required for OpenVMS AXP Version 6.1.

Creating an OpenVMS AXP Step 2 Device Driver from an OpenVMS VAX Device Driver

Provides information for migrating a device driver used on OpenVMS VAX to a Step 2 device driver used on OpenVMS AXP Version 6.1.

Guide to OpenVMS AXP Performance Management

Introduces and explains the techniques used to optimize performance on an OpenVMS Alpha system.

Guide to OpenVMS Performance Management

Introduces and explains the techniques used to optimize performance on an OpenVMS VAX system.

Migrating an Application from OpenVMS VAX to OpenVMS Alpha

Describes how to create an OpenVMS Alpha version of an OpenVMS VAX application. Provides an overview of the VAX to Alpha migration process and information to help you plan a migration. It discusses the decisions you must make in planning a migration and the ways to get the information you need to make those decisions. In addition, this manual describes the migration methods available so that you can estimate the amount of work required for each method and select the method best suited to a given application.

Migrating an Environment from OpenVMS VAX to OpenVMS Alpha

Describes how to migrate a computing environment from an OpenVMS VAX system to an OpenVMS Alpha system or a mixed-architecture cluster. Provides an overview of the VAX to Alpha migration process and describes the differences in system and network management on VAX and Alpha computers.

Migrating to an OpenVMS AXP System: Planning for Migration

Describes the general characteristics of RISC architectures, compares the Alpha architecture to the VAX architecture, and presents an overview of the migration process and a summary of migration tools provided by Compaq. The information in this manual is intended to help you define the optimal migration strategy for your application.

Migrating to an OpenVMS AXP System: Recompiling and Relinking Applications

Provides detailed technical information for programmers who must migrate high-level language applications to OpenVMS Alpha. Describes how to set up a development environment to facilitate the migration of applications, helps programmers identify application dependencies on elements of the VAX architecture, and introduces compiler features that help resolve these dependencies. Individual sections of this manual discuss specific application dependencies on VAX architectural features, data porting issues (such as alignment concerns), and the process of migrating VAX shareable images.

OpenVMS Alpha Guide to 64-Bit Addressing and VLM Features

Introduces and describes OpenVMS Alpha operating system support for 64-bit virtual addressing and Very Large Memory (VLM). Intended for system and application programmers, this guide highlights the features and benefits of OpenVMS Alpha 64-bit and VLM capabilities. It also describes how to use these features to enhance application programs to support 64-bit addresses and to efficiently harness very large physical memory.

OpenVMS Alpha System Dump Analyzer Utility Manual

Explains how to use the System Dump Analyzer utility to investigate system failures and examine a running OpenVMS Alpha system. Alpha programmers should refer to this manual; VAX programmers should refer to the *OpenVMS VAX System Dump Analyzer Utility Manual*.

OpenVMS AXP Device Support: Developer's Guide

Describes how to write a driver for OpenVMS Alpha for a device not supplied by Compaq.

OpenVMS AXP Device Support: Reference

Provides the reference material for the *Writing OpenVMS Alpha Device Drivers in C* by describing the data structures, macros, and routines used in device-driver programming.

OpenVMS Bad Block Locator Utility Manual

Describes how to use the Bad Block Locator utility to locate bad blocks on older types of media.

OpenVMS Compatibility Between VAX and Alpha

Compares and contrasts OpenVMS on VAX and Alpha computers, focusing on the features provided to end users, system managers, and programmers.

OpenVMS Developer's Guide to VMSINSTAL

Describes the VMSINSTAL command procedure and provides guidelines for designing installation procedures that conform to standards recommended by Compaq. Intended for developers who are designing installation procedures for software products layered on the OpenVMS operating system.

OpenVMS DIGITAL Standard Runoff Reference Manual

Describes the DSR text-formatting utility.

OpenVMS EDT Reference Manual

Contains complete reference information for the EDT editor.

OpenVMS Exchange Utility Manual

Describes how to use the Exchange utility to transfer files between some foreign format volumes and OpenVMS native volumes.

OpenVMS Glossary

Defines terms specific to OpenVMS that are used throughout the documentation.

OpenVMS National Character Set Utility Manual

Describes how to use the National character set utility to build NCS definition files.

OpenVMS Obsolete Features Manual

Presents the DCL commands, system services, RTL routines, and utilities made obsolete by VMS Version 4.0 through Version 5.0. Includes an appendix of DCL commands, RTL routines, and utilities eliminated from VMS Version 4.0.

OpenVMS Programming Environment Manual

Provides a general description of Compaq products and tools that define the programming environment. Introduces facilities and tools such as the compilers, the linker, the debugger, the System Dump Analyzer, system services, and routine libraries.

OpenVMS Programming Interfaces: Calling a System Routine

Describes the OpenVMS programming interface and defines the standard conventions to call an OpenVMS system routine from a user procedure. The Alpha and VAX data type implementations for various high-level languages are also presented in this manual.

OpenVMS RTL DECtalk (DTK$) Manual

Documents the DECtalk support routines contained in the DTK$ facility of the OpenVMS Run-Time Library.

OpenVMS RTL Parallel Processing (PPL$) Manual

Documents the parallel-processing routines contained in the PPL$ facility of the OpenVMS Run-Time Library. Indicates which routines are specific to Alpha or VAX, as well as how routines function differently on each system.

OpenVMS Software Overview

Provides an overview of the OpenVMS operating system and some of its available products.

OpenVMS SUMSLP Utility Manual

Describes how to use the SUMSLP batch-oriented editor to update source files.

OpenVMS System Messages and Recovery Procedures Reference Manual

Contains an alphabetical listing of the errors, warnings, and informational messages issued by the operating system. Also provides the meaning of each message and a statement of the action to be taken in response to each message. This manual is in two volumes.

OpenVMS Terminal Fallback Utility Manual

Describes how to use the Terminal Fallback utility to manage the libraries, character conversion tables, and terminal parameters that are available within this utility.

OpenVMS VAX Card Reader, Line Printer, and LPA11–K I/O User's Reference Manual

Describes the card reader, laboratory peripheral accelerator, and line printer drivers on OpenVMS VAX.

OpenVMS VAX Device Support Manual

Describes how to write an OpenVMS VAX driver for a device not supplied by Compaq.

OpenVMS VAX Device Support Reference Manual

Provides the reference material for the *OpenVMS VAX Device Support Manual* by describing the data structures, macros, and routines used in device-driver programming.

OpenVMS VAX Patch Utility Manual

Describes how to use the Patch utility to examine and modify executable and shareable OpenVMS VAX images.

OpenVMS Wide Area Network I/O User's Reference Manual

Describes the DMC11/DMR11, DMP11 and DMF32, DR11-W and DRV11-WA, DR32, and asynchronous DDCMP interface drivers on OpenVMS VAX.

PDP–11 TECO User's Guide

Describes the operating procedures for the PDP-11 TECO (Text Editor and Corrector) program.

POLYCENTER Software Installation Utility User's Guide

Provides information on the POLYCENTER Software Installation utility, a new component that lets you install and manage software products that are compatible with the utility.

TCP/IP Networking on OpenVMS Systems

Provides an introductory overview of TCP/IP networking and describes OpenVMS DCL support for TCP/IP capabilities.

Index

A

Advanced Server for OpenVMS, 6–2
ANALYZE/PROCESS_DUMP command, 5–9
ATM
  cluster interconnect, 4–10
Attunity Connect "On Platform" Package, 2–3
Availability Manager, 6–1

B

Backup using Volume Shadowing for OpenVMS, 4–20
64-bit addresses
  support for, 5–5

C

Case preservation in file names, C RTL, 5–3
Class scheduler
  CPU scheduling, 4–2
Cluster interconnects
  LANs as cluster interconects, 4–11
Clusterwide intrusion detection, 4–8
Compaq 3X-DAPBA-FA adapter, 5–2
Compaq 3X-DAPCA-FA adapter, 5–2
Compaq Analyze, 6–6
Compaq BridgeWorks, 2–3
Compaq COM for OpenVMS, 2–2
Compaq Crash Analysis Tool (CCAT), 6–6
Compaq C RTL, 5–2
  case preservation in file names, 5–3
  exact case `argv` arguments, 5–4
  long file names as arguments, 5–3
  nested directory limitation lifted, 5–3
  new functions, 5–5
  shared access, 5–4
  `strptime` function, 5–2
  support for extended file specifications, 5–3
  UNIX file-spec translation, 5–5
Compaq DECwindows Motif, 6–2
Compaq Enterprise Directory Services for e-Business, 2–3
Compaq Fast Virtual Machine (Fast VM) for the Java 2 Platform, 2–2
Compaq Java 2 SDK, Standard Edition, for OpenVMS Alpha, 2–2
Compaq Secure Web Server for OpenVMS Alpha, 2–2
Compaq System Tools CD-ROM, 6–6
Compaq Volume Shadowing for OpenVMS, 4–20
Compaq XML Technology, 2–2
CPU Online Replace, 4–2
CPU scheduling
  class scheduler, 4–2

D

Daylight Savings Time
  automatic setting, 4–2
DCL command
  ANALYZE/IMAGE, 3–1
  ANALYZE/OBJECT, 3–1
  ANALYZE/PROCESS, 3–1
  DELETE, 3–1
  DELETE/INTRUSION, 3–1
  DIRECTORY, 3–1
  DISMOUNT, 3–1
  DUMP, 3–1
  INITIALIZE, 3–1
  MOUNT, 3–1
  SET AUDIT, 3–2
  SET CACHE, 3–2
  SET DEVICE, 3–2
  SET DISPLAY, 3–2
  SET FILE, 3–2
  SET PROCESS, 3–2
  SET RMS_DEFAULT, 3–2
  SET RMS_DEFAULT/QUERY_LOCK (Alpha), 5–14
  SET SERVER, 3–2
  SET VOLUME, 3–2
  SHOW CPU, 3–2
  SHOW DEVICES, 3–2
  SHOW INTRUSION, 3–2
  SHOW LICENSE, 3–2
  SHOW MEMORY, 3–2
  SHOW RMS_DEFAULT, 3–2
  SHOW SERVER, 3–2
  UNLOCK, 3–2

DCL lexical
F$GETDVI, 3–3
F$GETJPI, 3–3
F$GETQUI, 3–3
F$GETSYI, 3–3
Debugger multiprocess functionality, 5–7
DECevent, 6–6
Dedicated CPU Lock Manager, 4–4
enabling, 4–4
implementing, 4–4
setting affinity, 4–5
supporting fast path devices, 4–5
DISMOUNT/FORCE_REMOVAL command, 4–21
DSGGB-AA switch, 4–9
DSGGB-AB switch, 4–9
DUMP command, 5–17
Dump files
analyzing, 5–8
on different systems, 5–10
forcing, 5–10

E

ECP Collector, 6–4
ECP Performance Analyzer, 6–4
Enterprise Capacity and Performance (ECP), 6–4
/ERASE qualifier, 4–25
Exact case `argv` arguments, C RTL, 5–4
Executive
nonpaged design, 1–7
Extended File Cache, 4–7, 4–12
Extended file specifications
Compaq C RTL, 5–3

F

FAB$B_SHR field, 5–14
Fibre Channel support, 4–9
Fibre Channel tape support, 4–10
Firmware, 1–3
Floppy disks
serving, 4–9

G

Gigabit Ethernet
cluster interconnect, 4–10
GLX_SHM_REG system parameter, 4–15

H

HSG60 storage controller, 4–9

I

INITIALIZE command
/ERASE qualifier, 4–25
/SHADOW qualifier, 4–25
Intrusion detection
clusterwide, 4–8
IO_PREFER_CPUS system parameter, 4–6

K

Kerberos, 6–5
KGPSA-CA host adapter, 4–9

L

LANs
switched, 4–10
LCKMGR_CPUID system parameter, 4–5
LCKMGR_MODE system parameter, 4–4
LCKMGR_SERVER process, 4–4, 4–5
LDAP, 6–5
Lightweight Directory Access Protocol (LDAP), 6–5
Long file names
arguments to C RTL functions, 5–3

M

MA6000 storage subsystem, 4–9
MDR (Modular Data Router), 4–10
Minicopy operation, 4–20
write bitmaps, 4–20
MONITOR utility
new features, 4–7
Multiprocess debugging, 5–7

N

Nested directory limitation lifted, C RTL, 5–3
Network authentication
Kerberos, 6–5
Nonpaged executive change, 1–7

O

OpenVMS Cluster systems, 4–10
disaster-tolerant support, 4–21
Fibre Channel support, 4–9
floppy disk serving, 4–9
mixed-architecture support, 4–11
mixed-version support, 4–11
remedial kits, 4–12
switched LAN support, 4–10
warranted and migration support, 4–11

OpenVMS e-Business, 1–3, 2–1
OpenVMS Enterprise Directory for e-Business
LDAP, 4–6
X.500, 4–6
OpenVMS Registry, 5–16

P

Page-file sections
larger limits (Alpha), 5–5
PATHWORKS for OpenVMS, 6–6
PEDRIVER, 4–10
POLYCENTER Software Installation Utility, 5–8
Process dumps
disk quota, 5–10
privileged users, 5–11
security, 5–10, 5–11
Process dump tools, 5–8
Programming features, 1–6

R

RAB$V_NLK option, 5–14
RAB$V_NODLCKBLK option, 5–15
RAB$V_NODLCKWT option, 5–15
RAB$V_NQL option, 5–14
RAB$V_RRL option, 5–14
RAB$W_ROP field, 5–14
RAB$W_ROP_2 field, 5–14, 5–15
RAD_SUPPORT system parameter, 4–17
Record locking with RMS
controlling deadlock detection, 5–15
specifying no query record locking (Alpha), 5–14
using read-mode bucket locking (Alpha), 5–12
REG$CP, 5–16
Registry system service, 5–16
$REGISTRY system service, 5–16
Reliable Transaction Router (RTR), 2–3
REPEAT command
count parameter, 5–21
/UNTIL=condition qualifier, 5–21
Revision and Configuration Management (RCM) tools, 6–6
RMS
controlling deadlock detection, 5–15
minimizing locking for shared access to global buffers (Alpha), 5–13
minimizing record locking for read accesses to shared files (Alpha only), 5–13
setting global buffers on a data file, 5–13
specifying no query record locking (Alpha), 5–14

S

SCA Control Program, 4–11
SCACP
See SCA Control Program
SDA commands, 5–9, 5–17
Spinlock Tracing utility, 5–24
SDA qualifiers and parameters, 5–17
SDA Spinlock Tracing utility
SPL LOAD command, 5–24
SPL SHOW TRACE command, 5–25
SPL START COLLECT command, 5–25
SPL START TRACE command, 5–26
SPL STOP COLLECT command, 5–26
SPL STOP TRACE command, 5–26
SPL UNLOAD command, 5–26
SEARCH command
/STEPS qualifier, 5–21
Security
clusterwide intrusion detection, 4–8
Kerberos, 6–5
SET DEVICE command
new volume shadowing qualifiers, 4–21
site qualfiers, 4–22
SET OUTPUT command
/HEADER/NOHEADER qualifiers, 5–21
SET PROCESS command
/DUMP qualifier, 5–10
/NEXT qualifier, 5–21
SET RMS_DEFAULT/QUERY_LOCK command (Alpha), 5–14
SET SYMBOLIZE command, 5–18
/SHADOW qualifier, 4–25
Shared access, C RTL, 5–4
SHOW DEVICE command
/CDT qualifier, 5–22
/PDT qualifier, 5–22
/UCB qualifier, 5–22
SHOW GCT
/CHILDREN qualifier, 5–22
SHOW LOCK command
DPC keyword, 5–22
SHOW MEMORY command, 5–19
SHOW PFN_DATA command
/RAD [=n] qualifier, 5–22
SHOW POOL command
/BRIEF qualifier, 5–22
/CHECK qualifier, 5–22
/MAXIMUM_BYTES qualifier, 5–22
/STATISTICS [=ALL] qualifier, 5–22
/UNUSED qualifier, 5–22
SHOW PROCESS command
/FID_ONLY qualifier, 5–23
/GSTX=index qualifier, 5–23
/IMAGES [=ALL] qualifier, 5–23
/NEXT qualifier, 5–23
/PST qualifier, 5–23

SHOW RAD command, 5–19
SHOW RESOURCE command
  /OWNED qualifier, 5–24
SHOW SPINLOCKS command
  /COUNTS qualifier, 5–24
SHOW SUMMARY command
  /PROCESS_NAME=*process_name* qualifier, 5–24
SHOW TQE command, 5–20
`strptime` function, 5–2
System management features, 1–3
System parameters
  AUTO_DLIGHT_SAV, 4–15
  FAST_PATH_PORTS, 4–15
  LCKMGR_CPUID, 4–15
  LCKMGR_MODE, 4–16
  NPAGECALC, 4–16
  NPAGERAD, 4–16
  SHADOW_MAX_UNIT, 4–17
  VCC_MAX_IO_SIZE, 4–17
  VCC_READAHEAD, 4–17
  WBM_MSG_INT, 4–18
  WBM_MSG_LOWER, 4–18
  WBM_MSG_UPPER, 4–19
  WBM_OPCOM_LVL, 4–19
System services
  $CHECK_PRIVILEGES, 5–27
  $CLRAST, 5–27
  $DCLEXH, 5–27
  $DELETE_INTRUSION, 5–27
  $DEVICE_PATH_SCAN, 5–27
  $DISMOU, 5–27
  $EXPREG, 5–27
  $GETDVI, 5–27
  $GETJPI, 5–27
  $GETQUI, 5–27
  $GETRMI, 5–27
  $GETSYI, 5–27
  $IO_PERFORM, 5–27
  $MGBLSC, 5–27
  $MOUNT, 5–27
  $PERSONA_QUERY, 5–27
  $PROCESS_SCAN, 5–28
  $REGISTRY, 5–28
  $SCAN_INTRUSION, 5–28
  $SCHED, 5–28
  $SET_DEVICE, 5–28
  $SET_SECURITY, 5–28
  $SET_SYSTEM_EVENT, 5–28
  $SHOW_INTRUSION, 5–28
  $WAKE, 5–28

T

TCP/IP files
  VAX SDA, 5–28
TCPIP$BGDRIVER.STB
  global symbols, 5–29
TCPIP$INTEETACP.STB
  global symbols, 5–29
TCPIP$INTERNET_SERVICES.STB
  global symbols, 5–29
TCPIP$NET_GLOBALS.STB file, 5–28
TCPIP$NFS_GLOBALS.STB file, 5–28
TCPIP$NFS_SERVICES.STB file, 5–29
TCPIP$PROXY_GLOBALS.STB file, 5–28
TCPIP$PROXY_SERVICES.STB file, 5–29
TCPIP$PWIPACP.STB
  global symbols, 5–29
TCPIP$PWIPDRIVER.STB
  global symbols, 5–29
TCPIP$PWIP_GLOBALS.STB file, 5–28
TCPIP$TNDRIVER.STB
  global symbols, 5–29
TCPIP$TN_GLOBALS.STB file, 5–28

U

UNDEFINE command, 5–20
UNIX file-spec translation, C RTL, 5–5
User features, 1–3
Utility routines
  online help, 3–3

V

VAX SDA
  TCP/IP files, 5–28
Visual Threads
  new version support, 5–29
Volume Shadowing for OpenVMS
  disaster-tolerant support, 4–21
  minicopy, 4–12, 4–20

W

Web-Based Enterprise Services (WEBES), 6–6
World-Wide PostScript Printing Subsystem, 3–4
Write bitmaps, 4–20
WWPPS, 3–4

X

XFC
  See Extended File Cache