OpenVMS Version 7.2 New Features Manual

Order Number: AA–QSBFC–TE

January 1999

This manual describes the new features associated with the OpenVMS Alpha and OpenVMS VAX Version 7.2 operating systems.

Revision/Update Information: This manual supersedes the *OpenVMS Version 7.1 New Features Manual*

Software Version: OpenVMS Alpha Version 7.2
OpenVMS VAX Version 7.2

Compaq Computer Corporation
Houston, Texas

January 1999

ZK6520

The OpenVMS documentation set is available on CD–ROM.

Contents

4 Programming Features

5 Compaq Galaxy Software Architecture on OpenVMS Alpha

Figures

Tables

Preface

Intended Audience

This manual is intended for general users, system managers, and programmers who use the OpenVMS operating system.

This document describes the new features related to Version 7.2 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 7.2.

Document Structure

This manual is organized as follows:

- Chapter 1 contains a summary of the new OpenVMS software features.
- Chapter 2 describes new features of interest to general users of the OpenVMS VAX and OpenVMS Alpha operating systems.
- Chapter 3 describes new features that are applicable to the tasks performed by system managers.
- Chapter 4 describes new features that support programming tasks.
- Chapter 5 provides an overview of the Compaq Galaxy Software Architecture on OpenVMS Alpha.
- The appendix lists the new messages for OpenVMS Version 7.2 from the Help Message database.

Related Documents

For additional information on Open Systems Software Group (OSSG) products and services, access the following OpenVMS World Wide Web address:

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

Reader's Comments

Compaq welcomes your comments on this manual.

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

Internet	**openvmsdoc@zko.mts.dec.com**
Fax	603 884-0120, Attention: OSSG Documentation, ZKO3-4/U08

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.digital.com:81/`

If you need help deciding which documentation best meets your needs, call 800-DIGITAL (800-344-4825).

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.

1

Summary of OpenVMS Version 7.2 New Features

OpenVMS Version 7.2 leverages existing capabilities and provides significant new functionality and features. Some of the most significant new OpenVMS Version 7.2 features include:

- Compaq Galaxy Software Architecture on OpenVMS Alpha
- Seamless integration between OpenVMS and Windows NT
- Premier OpenVMS Clustering—includes increased cluster performance for large-scale databases and easier system management with cluster-wide logical names
- Java Development Kit
- Extended File Specifications support
- TCP/IP Services for OpenVMS—new and improved. Based on the UNIX kernel
- Enhanced system management
- New network support for Gigabit Ethernet
- Base infrastructure support for Fibre Channel as a storage interconnect
- Ongoing improvements—to boost operating system performance, reliability, and availability
- Year 2000 readiness

OpenVMS Version 7.2 includes all the capabilities of OpenVMS Version 7.1 and the OpenVMS Version 7.1-1Hx hardware releases.

Table 1–1 summarizes each feature provided by OpenVMS Alpha and OpenVMS VAX Version 7.2 and presents these features according to their functional component (general user, system management, or programming).

Table 1–1 Summary of OpenVMS VAX and OpenVMS Alpha Version 7.2 Software Features

General User Features	
DCL Commands and Qualifiers	A new DCL command has been added: • SET PREFERRED_PATH—Provides a command line interface to the preferred path QIO function. In addition, changes have been made to several DCL commands and qualifiers to support Extended File Specifications, Hierarchical Storage Management (HSM), and clusterwide logical names.
Documentation	Three new OpenVMS manuals have been added to the documentation set: the *OpenVMS Alpha Galaxy Guide*, the *Guide to OpenVMS Extended File Specifications*, and the *OpenVMS Alpha System Analysis Tool Manual*. For Version 7.2, the OpenVMS Documentation CD–ROM supports a dual OpenVMS and Windows-based PC format.
MIME Interpretation and Composition Editor	The MIME Interpretation and Composition Editor utility (MICE) allows you to read and compose MIME-encoded mail messages.
System Management Features	
BACKUP utility	The OpenVMS Version 7.2 BACKUP utility contains the following improvements: • A new BACKUP qualifier (/NOINCREMENTAL) allows you to back up modified directories in one operation without additionally having to save all the files and subdirectories within them. • Improvements to BACKUP performance
Bitmap limits increased	Beginning with OpenVMS Version 7.2, the limits of storage and index file bitmaps have been increased.
DECnet installation choices	Beginning with OpenVMS Version 7.2, the OpenVMS Alpha and OpenVMS VAX installation menus now include both DECnet-Plus and DECnet Phase IV as networking software choices. During the OpenVMS installation or upgrade, you are given the option to install either DECnet-Plus or DECnet Phase IV.
Deferred memory testing on AlphaServer 4100 computers	To speed up the time between system power-on and user login, the system manager can now defer a portion of memory testing on AlphaServer 4100 computers. This option limits the console to a minimum amount of memory testing and leaves the rest for the operating system to test.
DIGITAL Server MIB (DSM) subagents implemented on OpenVMS (Alpha only)	OpenVMS Alpha Version 7.2 implements the DSM subagents on the AlphaServer 800, 1000, 4000, 4100, 8200, and 8400 systems. With the DSM subagents, you can remotely determine and manage important information such as: Firmware revision numbers Base system descriptions FRU (field replaceable unit) information and descriptions Processor and cache status Interface configurations Environmental conditions in the system enclosure that might be detrimental to the hardware

(continued on next page)

Table 1–1 (Cont.) Summary of OpenVMS VAX and OpenVMS Alpha Version 7.2 Software Features

System Management Features	
Ethernet and Fast Ethernet Support (Alpha only)	OpenVMS Alpha Version 7.2 provides run-time and boot support for the DE500-BA PCI Ethernet network interface card (NIC) and for the DE500-FA PCI Fast Ethernet network interface card. The former supports a maximum distance of 100 meters; the latter supports a maximum distance of 2000 meters.
Extended File Specifications (Alpha only)	Extended File Specifications is a file handling environment that eliminates many of the file naming restrictions currently imposed by OpenVMS. This environment provides consistent file handling across both OpenVMS and Windows NT systems in a PATHWORKS environment.
Fast Skip for SCSI tape drives	There is a new qualifier for the SET MAGTAPE command, /FAST_SKIP, that allows you to skip by file mark or by record. The options are: PER_IO (default), ALWAYS, and NEVER.
FastTrack Web Server (Alpha only)	FastTrack Web Server for OpenVMS Alpha provides a way to create, build, publish, and serve Web pages and applications.
Fibre Channel support (Alpha only)	Fibre Channel is an ANSI standard network and storage interconnect that offers many advantages, including high speed transmission, long interconnect distances, and support of multiple protocols. OpenVMS Alpha will support Fibre Channel as a storage interconnect for single systems and as a shared storage interconnect in multihost OpenVMS Cluster systems. This support will become available shortly after the release of OpenVMS Version 7.2.
Firmware revision checking (Alpha only)	OpenVMS Alpha Version 7.2 provides enhanced firmware checking for systems during a boot operation. When you boot the OpenVMS Alpha operating system CD–ROM, the system now automatically checks the version of console firmware (including PALcode) that is running on your computer.
InfoServer Service Listing (ISL)	OpenVMS Version 7.2 offers system managers a quicker way to view and select services from a Compaq InfoServer. Service information is now grouped by server and displayed alphabetically.
LANCP Commands to Manage Classical IP over ATM (Alpha only)	On Alpha systems, Define Device and Set Device have a new qualifier for Classical IP, and a new keyword for the /CLIP qualifier.
MONITOR utility: TCP/IP support added	The Monitor utility has been enhanced to be able to use either TCP/IP (if available) or DECnet as its transport.

(continued on next page)

Table 1–1 (Cont.) Summary of OpenVMS VAX and OpenVMS Alpha Version 7.2 Software Features

System Management Features		
OpenVMS Cluster Systems	The following are new OpenVMS Cluster features:	
	New CIPCA adapter support	A new CIPCA adapter has been qualified since the release of OpenVMS Version 7.1.
	Clusterwide logical names	Clusterwide logical name support includes new attributes for the $TRNLNM and $GETSYI system services and a change in behavior for the $CRELNT system service which allows easier system management of an OpenVMS Cluster.
	Gigabit Ethernet as a cluster interconnect (Alpha only)	Gigabit Ethernet is supported as an OpenVMS Cluster interconnect for OpenVMS Version 7.1–2. The same support will be available on OpenVMS Version 7.2 soon after its release.
	Intra-cluster Communications (ICC)	The ICC system services are a new applications programming interface (API) for intra-cluster communications.
	Lock Manager performance improvements	The enhanced lock manager software improves performance of applications that issue a large number of lock manager requests. The enhancements also improve application scaling.
	MEMORY CHANNEL	MEMORY CHANNEL enhancements include support for larger configurations, new MEMORY CHANNEL hardware, and more robust performance.
	Multipath SCSI support (Alpha only)	Multipath SCSI support in OpenVMS Alpha provides failover from one path to a device to another path to the same device. Failover is provided between multiple direct SCSI paths. Failover between direct SCSI paths and MSCP served paths will be available soon after the release of OpenVMS Version 7.2. Multiple paths to the same device increase the availability of that device for I/O operations and increase performance in certain configurations. Multipath support is available for parallel SCSI configurations. Shortly after the release of OpenVMS Version 7.2, multipath support will also be available for Fibre Channel configurations.
	SCSI OpenVMS Cluster system supports four nodes	With the introduction of the SCSI hub DWZZH-05, the maximum number of nodes in a SCSI multihost OpenVMS Cluster system has been raised from three to four.
	Ultra SCSI configuration support	OpenVMS Version 7.2 supports Ultra SCSI operations in single-host and multihost configurations.
	Warranted and migration support	OpenVMS Alpha Version 7.2 and OpenVMS VAX Version 7.2 provide two levels of support, warranted and migration, for mixed-version and mixed-architecture OpenVMS Cluster systems.

(continued on next page)

Table 1–1 (Cont.) Summary of OpenVMS VAX and OpenVMS Alpha Version 7.2 Software Features

System Management Features	
OpenVMS Management Station	OpenVMS Management Station makes it easy to manage disk storage devices across multiple OpenVMS Cluster systems.
OpenVMS Registry	To allow OpenVMS and Windows NT to interoperate, Compaq has provided a registry on OpenVMS.
PCI Gigabit Ethernet support (Alpha only)	OpenVMS Alpha Version 7.2 provides run-time support for the Digital PCI-to-Gigabit Ethernet adapter. This adapter incorporates a new technology that transfers data at a rate of one gigabit per second—ten times the rate of a Fast Ethernet adapter. Gigabit Ethernet technology addresses congestion experienced at the backbone and server levels by today's networks.
PCSI qualifiers	Beginning with OpenVMS Version 7.2, the POLYCENTER Software Installation (PCSI) utility has two new qualifiers that you can use with a number of PRODUCT commands.
RMS CONVERT Utility Enhancements	The CONVERT utility has many enhancements for OpenVMS Version 7.2.
SYSMAN RESERVED_ MEMORY	New and modified subcommands and qualifiers have been added to reserve a block of contiguous page frame numbers (PFNs).
System parameters	OpenVMS Alpha Version 7.2 introduces several new system parameters.
Tape density support enhanced (Alpha only)	Enhanced tape density support permits the selection of any density and any compression supported by a tape drive.
TCP/IP services	With OpenVMS Version 7.2, the DIGITAL TCP/IP Services for OpenVMS Version 5.0 product replaces Version 4.2 (also known as UCX). Version 5.0 completes the change initiated several releases ago when the product name changed from ULTRIX Connection (UCX) to DIGITAL TCP/IP Services for OpenVMS. Several new TCP/IP features, including a new kernel and support for the gateway routing (GATED) daemon, are available in this release.
Year 2000 readiness	OpenVMS Version 7.2 includes all the Year 2000 enhancements that shipped in the Year 2000 enhancement kits for OpenVMS Version 7.1 and Version 7.1–1H1. These Year 2000 enhancements are the result of a rigorous and comprehensive analysis of the entire OpenVMS operating system, including extensive OpenVMS testing.

(continued on next page)

Table 1–1 (Cont.) Summary of OpenVMS VAX and OpenVMS Alpha Version 7.2 Software Features

Programming Features	
Additional network adapters (Alpha only)	OpenVMS Alpha Version 7.2 offers support for the following new adapters: ATMWORKS 351 adapter (for communication over an ATM network) DE500-BA and DE504-BA adapters (for communication over a PCI Ethernet network) DE500-FA adapter (for communication over a PCI FastEthernet network) DEGPA-SA adapter (Alpha only—for communication over a PCI Gigabit Ethernet Network)
Clusterwide logical name support	Clusterwide logical name support includes new attributes for the $TRNLNM and $GETSYI system services and a change in behavior for the $CRELNT system service.
COM for OpenVMS	COM (Component Object Model) is a technology from Microsoft that allows developers to create distributed network objects. DIGITAL and Microsoft jointly developed the COM specification. First released as NetOLE (Network Object Linking and Embedding) and then renamed DCOM (Distributed COM), the COM specification now includes network objects. COM for OpenVMS is an implementation of the Microsoft code that supports the COM draft standards.
Common file qualifier routines	New common file qualifier routines (UTIL$CQUAL) allow you to parse the command line for qualifiers related to certain file attributes, and to match files you are processing against the selected criteria retrieved from the command line.
DECthreads	DECthreads for OpenVMS Version 7.2 contains the following new features: • Yellow zone stack overflow—area of memory that enables applications to catch stack overflow conditions and attempt recovery or terminate gracefully • Read-write locks—allows many threads to have simultaneous read-only access to data while allowing only one thread to have write access • Improved debugging support
DIGITAL DCE Remote Procedure Call (RPC)	Beginning with OpenVMS Version 7.2, Remote Procedure Call (RPC) functionality is integrated into the operating system. RPC provides connectivity between individual procedures in an application across heterogeneous systems in a transparent way. Using RPC, an application can interoperate with either DIGITAL DCE or Microsoft RPC applications.
Fast I/O and buffer objects for global sections (Alpha only)	As of OpenVMS Alpha Version 7.2, VLM applications can use Fast I/O for memory shared by processes through global sections. In prior versions of OpenVMS Alpha, buffer objects could be created only for process private virtual address space.
Fast Skip for SCSI tape drives	If you access your tape drive via your own user-written program, you can use a new modifier, IO$M_ALLOWFAST, to control the behavior of the IO$_SKIPFILE function.

(continued on next page)

Table 1–1 (Cont.) Summary of OpenVMS VAX and OpenVMS Alpha Version 7.2 Software Features

Programming Features	
Fast Path support (Alpha only)	Beginning with OpenVMS Alpha Version 7.1, Fast Path supports disk I/O for the CIXCD and the CIPCA ports.
High-Performance Sort/Merge Utility (Alpha only)	The high-performance Sort/Merge utility for OpenVMS Version 7.2 supports the following: • Threads • Indexed sequential output file organization • Output file overlay • Statistical summary information
Intra-Cluster Communications	Intra-cluster communications (ICC) provides new system services that form a supported applications programming interface (API) to the existing interprocess communications services (IPC). This allows application program developers to create connections between different processes on the same or different systems within a single OpenVMS Cluster.
Java Development Kit (Alpha only)	The Java Development Kit is now a standard component of the OpenVMS operating system. This kit can be used to develop and run Java applets and programs on OpenVMS Alpha systems.
Kernel threads enhancements (Alpha only)	Kernel threads enhancements include the increased number of threads per-process, the ability to change the base priorities of all kernel threads with one call, and detection of thread stack overflow.
OpenVMS Debugger enhancements	The OpenVMS Debugger includes the following features with OpenVMS Version 7.2: • New client/server interface—This new interface allows you to use Windows-based client applications to debug programs that reside and run on OpenVMS VAX or Alpha systems. • Support for C++ Version 5.5 or later (Alpha only)—OpenVMS Version 7.2 provides you with the ability to debug C++ programs with the OpenVMS Alpha Debugger. • New DECthreads command—When debugging programs that use DECthreads Version 3.13 or greater, you can directly access the DECthreads debugger with the PTHREAD command.
OpenVMS RTL Routine LIB$GET_LOGICAL	For OpenVMS Version 7.2, the RTL Routine LIB$GET_LOGICAL provides streamlined access to the $TRNLNM system service.

(continued on next page)

Table 1–1 (Cont.) Summary of OpenVMS VAX and OpenVMS Alpha Version 7.2 Software Features

Programming Features	
Record Management Services enhancements	RMS includes the following enhancements for OpenVMS Version 7.2: • Support for Coordinated Universal Time (UTC) format—RMS and FAL now support the 128-bit UTC format for the exchange of date and time information about files. • Support for a new file length hint attribute—RMS under some conditions now maintains file length hint data (record count and user data byte count) that may be retrieved by users using an item list XAB. • The Analyze/RMS_File Utility has a new qualifier: /UPDATE_HEADER. • The XAB$_UCHAR_PRESHELVED item for file user characteristics for item list XAB (XABITM) of RMS is available. • Global buffer performance enhancement—RMS global buffer hashing replaces the sequential buffer lookup with a hashing scheme that improves performance for lookups in the global cache.
Scheduling processes with soft affinity (Alpha only)	In OpenVMS Alpha Version 7.2, the scheduling algorithm for soft affinity has been improved.
Security features (Alpha only)	OpenVMS Alpha Version 7.2 contains the following security changes: Persona support — Includes changes and additions to the $PERSONA system services to support Windows NT-style security semantics. Per-thread security — Allows each thread of execution within a multithreaded process to have an individual security profile.
Shared address data (Alpha only)	Using shared address data saves physical memory by reducing fixup CPU time and I/O for fixup during image activation. Performance is improved because of the reduced demand on physical memory with shared images.
System Dump Analyzer enhancements (Alpha only)	SDA includes the following enhancements for OpenVMS Version 7.2: • Enhanced SDA commands—Many new qualifiers were added to the SHOW and VALIDATE commands. • SDA CLUE Dump Off System Disk—SDA CLUE allows the system dump file to be written to a device other than the system disk. • SDA extensions—OpenVMS Alpha SDA extensions expand the functionality of the System Dump Analyzer with the addition of new routines.
System services	Several new system services have been added and others have been modified to support OpenVMS Version 7.2.

(continued on next page)

Table 1–1 (Cont.) Summary of OpenVMS VAX and OpenVMS Alpha Version 7.2 Software Features

Compaq Galaxy Software Architecture on OpenVMS Alpha	
Compaq Galaxy Software Architecture (Alpha only)	OpenVMS Alpha Version 7.2 introduces a model of computing that allows multiple instances of OpenVMS to execute cooperatively in a single computer. With Adaptive Partitioned Multiprocessing (APMP), many processors (and other physical resources) are partitioned in order to run multiple instances of operating systems.
OpenVMS System Messages	
New system messages	This appendix alphabetically lists and describes new messages that can occur before the system is fully functional; that is, when the Help Message database is not accessible.

2
General User Features

This chapter provides new features information for all users of the OpenVMS operating system.

2.1 DCL Commands, Qualifiers, and Lexical Functions

This section describes new and changed DCL commands, qualifiers, and lexical functions for OpenVMS Version 7.2.

2.1.1 New SET PATH_PREFERRED Command

OpenVMS Version 7.2 introduces the SET PATH_PREFERRED command as a command line interface to the QIO function for disk and tape drives. SET PATH_PREFERRED has the following qualifiers:

- /FORCE
- /HOST
- /NOHOST

For more information about SET PATH_PREFERRED and its qualifiers, refer to the *OpenVMS DCL Dictionary*.

2.1.2 Changed DCL Commands and Qualifiers

Several commands and qualifiers have been modified to support the Extended File Specifications, Hierarchical Storage Management, and clusterwide logical names. The following table contains a summary of these changes. For more information about changed commands and qualifiers, refer to the *OpenVMS DCL Dictionary*.

Table 2–1 Changed DCL Commands and Qualifiers

DCL Command	New Features
COPY	Support added for Extended File Specifications.
	• Added new qualifier, /STYLE, with new keywords, EXPANDED and CONDENSED.
DEBUG	Support added for CLIENT/SERVER.
	• Added new qualifiers, /CLIENT and /SERVER.

(continued on next page)

Table 2–1 (Cont.) Changed DCL Commands and Qualifiers

DCL Command	New Features
DELETE	Support added for Extended File Specifications. • Added new qualifier, /STYLE, with new keywords, EXPANDED and CONDENSED.
DIRECTORY	Support added for the Hierarchical Storage Management (HSM) project. • Added new keywords—PRESHELVED AND NOPRESHELVED—for /SELECT qualifier. • Added PRESHELVED display information to /SHELVED_STATE qualifier. Support added for Extended File Specifications. • Added new qualifier, /STYLE, with new keywords, EXPANDED and CONDENSED. • Added display item to /FULL to display client attributes.
DUMP	Support added for Extended File Specifications. • Added display item to /DIRECTORY to display Name type attribute. • Added display item to /HEADER to display new attributes. • Added new qualifier, /STYLE, with new keywords, EXPANDED and CONDENSED.
EXCHANGE NETWORK	Support added for Extended File Specifications. • Added new qualifier, /STYLE, with new keywords, EXPANDED and CONDENSED.
INITIALIZE	Support added for Extended File Specifications. • Added a new qualifier, /STRUCTURE=5. Support added for DECram allocation increase. • Added new information to /SIZE qualifier for DECram size allocation increase.
Java	Support added for Java Development Kit (JDK). • New DCL command and qualifiers added.
MIME	Support added for the MIME utility.
PRINT	Support added for Extended File Specifications. • Added new qualifier, /STYLE, with new keywords, EXPANDED and CONDENSED.

(continued on next page)

Table 2–1 (Cont.) Changed DCL Commands and Qualifiers

DCL Command	New Features
PURGE	Support added for Extended File Specifications. • Added new qualifier, /STYLE, with new keywords, EXPANDED and CONDENSED.
RENAME	Support added for Extended File Specifications. • Added new qualifier, /STYLE, with new keywords, EXPANDED and CONDENSED.
SEARCH	Support added for Extended File Specifications. • Added new qualifier, /STYLE, with new keywords, EXPANDED and CONDENSED.
SET DEFAULT	Support added for Extended File Specifications. • Modified the directory-spec item. • Added an example for the /PARSE_STYLE qualifier.
SET DEVICE	Support added for Fast Path. Support added for multipath parallel SCSI and Fibre Channel configurations. • Added new qualifiers, /PATH, /ENABLE, /NOENABLE, /POLL, /NOPOLL, /SWITCH.
SET DIRECTORY	Support added for Extended File Specifications. • Added new qualifier, /STYLE, with new keywords, EXPANDED and CONDENSED.
SET FILE	Support added for Extended File Specifications. • Added new qualifier, /STYLE, with new keywords, EXPANDED and CONDENSED.
SET MAGTAPE	Support added for Fast Skip. • Added new qualifier, /FAST_SKIP, with new keywords, PER_IO, ALWAYS, and NEVER.
SET PROCESS	Support added for Extended File Specifications. • Added a new qualifier, /PARSE_STYLE=(keyword), where keywords are TRADITIONAL and EXTENDED.

(continued on next page)

Table 2–1 (Cont.) Changed DCL Commands and Qualifiers

DCL Command	New Features
SET SECURITY	Support added for Extended File Specifications. • Added new qualifier, /STYLE, with new keywords, EXPANDED and CONDENSED. Support added for Intra-Cluster Communications (ICC). • Added new Object Class, ICC-ASSOCIATION.
SET VOLUME	Support added for Extended File Specifications. • Added a new qualifier, /STRUCTURE_LEVEL=5 device-name[:][,...].
SHOW CPU	Support added for Fast Path.
SHOW DEVICES	Support added for Multipath. • Added new qualifier, /MULTIPATH.
SHOW DEVICES/FULL	Support added for Fast Path. Support added for Extended File Specifications. • Updated the display information to show the disk structure level.
SHOW MEMORY/POOL SHOW MEMORY/POOL /FULL	Support added for Lock Manager. • Added Lock Manager Dynamic Memory and Bus Addressable Memory attributes to SHOW MEMORY /POOL and SHOW MEMORY/POOL/FULL.
SHOW SECURITY	Support added for Intra-Cluster Communications (ICC). • Added new Object Class, ICC-ASSOCIATION.
SHOW SYSTEM	Support added for threads. • Added new qualifier, /MULTITHREAD.
SUBMIT	Support added for Extended File Specifications. • Added new qualifier, /STYLE, with new keywords, EXPANDED and CONDENSED.
TYPE	Support added for Extended File Specifications. • Added new qualifier, /STYLE, with new keywords, EXPANDED and CONDENSED.

2.1.3 New and Changed DCL Lexical Functions

The following table contains a summary of new and changed DCL lexical functions. Refer to the *OpenVMS DCL Dictionary* for more information.

Table 2–2 Summary of Changes to DCL Lexical Functions

Lexical Function	New Features
F$FILE_ATTRIBUTES	Support added for Extended File Specifications. • Added new item codes: FILE_LENGTH_HINT, VERLIMIT, DIRECTORY.
F$GETDVI	Support added for Extended File Specifications. • Added new attributes to the ACPTYPE item code.
F$GETENV	Support added for new lexical. • Returns the value of a specified console environment variable.
F$GETJPI	Support added for Extended File Specifications. • Added new item codes: PARSE_STYLE_PERM and PARSE_STYLE_IMAGE.
F$GETSYI	Support added for clusterwide logical names. • Added the CWLOGICALS item code. • Removed references that certain item codes are for local nodes only.
F$TRNLNM	Support added for clusterwide logical names. • Modified the case argument. • Added the CLUSTERWIDE item code. • Added two new examples.

2.2 Documentation Features

OpenVMS Version 7.2 includes some significant documentation on new features.

For this release, OpenVMS is shipping the following new manuals to support new and enhanced functionality:

- The *OpenVMS Alpha Galaxy Guide* introduces Compaq Galaxy Software Architecture concepts and describes how to create, manage, and use an OpenVMS Galaxy computing environment.
- The *Guide to OpenVMS 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.
- The *OpenVMS Alpha System Analysis Tool Manual* explains how to use various Alpha system analysis tools to investigate system failures and examine a running OpenVMS system. These tools are useful for the system programmer who must investigate the causes of system failures and debug kernel mode code. The tools available are system dump analysis (SDA), system-code debugger, system dump debugger, and watchpoint utility.

Version 7.2 of the *OpenVMS System Manager's Manual* contains the following new appendixes:

- Managing the OpenVMS Registry

  This appendix explains how to install and manage the OpenVMS Registry.

- DIGITAL MIB Subagents Implemented on OpenVMS Alpha

  The Extensible Simple Network Management Protocol (eSNMP) allows network managers to manage many different types of devices across all network and vendor boundaries through the use of databases called MIBs (Management Information Bases). This appendix describes the DIGITAL Server MIB and the DIGITAL Cluster MIB.

In addition, selected manuals from the integrated layered products and OpenVMS archived manuals are also included with the documentation.

For Version 7.2, the OpenVMS Documentation CD–ROM supports a dual OpenVMS and Windows-based PC format. This means that you can mount and read the CD–ROM on both OpenVMS VAX and Alpha systems and a Windows 95 or Windows NT system.

Table 2–3 provides information about the files and manuals that are provided on the two partitions.

Table 2–3 Documentation CD–ROM File Types

File Type	Platform	Book Type
HTML	OpenVMS/Windows	OpenVMS Manuals
PostScript	OpenVMS	OpenVMS Manuals Archived Manuals Layered Product Release Notes and Install Guides
Bookreader	OpenVMS	Archived manuals Selected layered product manuals
PDF	Windows	Archived manuals
Text	OpenVMS/Windows	Release Notes, New Features, Install Guides

For the first time, OpenVMS is also shipping the executable files for the following browsers and viewers:

- Netscape Navigator Version 3.03 for OpenVMS Alpha and VAX
- Adobe Acrobat Reader for reading PDF files
- Bookreader for OpenVMS Alpha and VAX

2.3 The MIME Utility

The MIME utility allows you to read and compose MIME-encoded mail messages. The standard used to attach nontext files to mail messages is the Multipurpose Internet Mail Extension (MIME). With MIME, nontext files, such as graphics or sound files, are encoded and sent as plain text, although that text may not be readable. The MIME utility decodes MIME files to their original form and allows you to create MIME-encoded files, which can be sent as mail messages using the OpenVMS Mail utility.

Complete instructions for using the MIME utility to decode or encode files are found in the *OpenVMS User's Manual*.

Information about the installation and administration of the MIME utility is found in the *OpenVMS System Manager's Manual*.

3
System Management Features

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

3.1 BACKUP Utility

The following section discusses improvements to the BACKUP utility for OpenVMS Version 7.2.

3.1.1 /NOINCREMENTAL Qualifier

On a save operation, you can use the new BACKUP qualifier /NOINCREMENTAL to avoid saving *all* the files and subdirectories under directories that have been modified. Use this qualifier only if you are sure that you want to save specific files and do not want to save all data.

Prior to OpenVMS Version 6.2, the system, by default, did not save the files and subdirectories that were under directories that had been modified. In OpenVMS Versions 7.0 and 7.1, to ensure a successful restore, the system saved all files and subdirectories under directories that had been modified. This behavior, however, sometimes resulted in saving files and subdirectories that were not needed for later restore operations. The /NOINCREMENTAL qualifier has been implemented to allow you more control over the amount of file data that is saved.

Usage Notes

- /NOINCREMENTAL has no meaning in a restore operation.
- /NOINCREMENTAL might be useful for the incremental operations on PATHWORKS servers that are discussed in the *OpenVMS System Manager's Manual: Essentials*.

3.1.2 Performance Improvements

Work on the BACKUP utility for this release has resulted in the following performance improvements:

- For a badly fragmented disk, a significant reduction in run time for an image backup—at times as much as 30% less time
- Decrease in overall CPU usage
- Decreased image size, especially the BACKUPSHR.EXE file

3.2 Bitmap Limits Increased

In previous versions of OpenVMS, both storage and index file bitmaps were limited to 255 blocks. This size, in turn, limited a volume to approximately one million allocation units, or clusters. Larger disks were required to have a larger cluster factor to accommodate the limit; for example, a 9 GB disk required a cluster factor of 18.

Beginning with OpenVMS Version 7.2, the limits of storage and index file bitmaps have been increased as follows:

Type of Bitmap	Limit
Storage bitmap	65535 blocks
Index file bitmap	4095 blocks

The increased bitmap limits have the following advantages:

- For the foreseeable future, they allow you to use space more efficiently with small files.
- They increase the number of files allowed on a volume to the architectural maximum of approximately 16 million.

The behaviors of the INITIALIZE and BACKUP commands reflect the larger bitmap sizes. On OpenVMS Version 7.1 and earlier versions, you cannot mount a volume with bitmaps larger than 255 blocks. For this reason, the default cluster factor that the INITIALIZE command chooses on Files–11 structure level 2 disks limits the storage bitmap to 255 blocks. However, now you can override the default with an explicit cluster factor to make the storage bitmap as large as you want.

In OpenVMS Version 7.2, for Files–11 structure level 5 volumes, the INITIALIZE command assigns a cluster factor of 3 to currently available disks. The default maximum number of files—and hence, the size of the index file bitmap—that the INITIALIZE command assigns increases to correspond to the size of the storage bitmap.

In OpenVMS Version 7.2, when you use the BACKUP/IMAGE command to restore or copy a volume, the BACKUP utility normally preserves the cluster factor. However, when you copy an ODS-2 volume whose storage bitmap is 255 blocks or smaller, if the output volume is larger, BACKUP will increase the cluster factor to keep the storage map 255 blocks or smaller. This behavior ensures that a volume that was usable on pre-Version 7.2 systems remains usable when copied to larger media. You can override the BACKUP behavior by setting the output volume's cluster factor to any size by initializing the output volume with the /CLUSTER qualifier and then invoking BACKUP with the /NOINITIALIZE qualifier.

Note

You cannot mount volumes with extended bitmaps on an OpenVMS Version 7.1 or earlier system.

3.3 DECnet Installation Choices

Beginning with OpenVMS Version 7.2, the OpenVMS Alpha and OpenVMS VAX operating system installation menus offer the choice to install either DECnet-Plus or DECnet Phase IV networking software. Note, however, that both DECnet products *cannot* run concurrently on your system. You must choose to install either DECnet-Plus or DECnet Phase IV. For DECnet Phase IV software service support, you must purchase an addendum to your standard support contract. See your local Compaq support representative for more information.

For detailed information about installing or upgrading DECnet-Plus and DECnet Phase IV, refer to the *OpenVMS Alpha Version 7.2 Upgrade and Installation Manual* and the *OpenVMS VAX Version 7.2 Upgrade and Installation Manual*.

3.4 Deferred Memory Testing on AlphaServer 4100 Computers

To speed up the time between system power-on and user login, the system manager can now defer a portion of memory testing on AlphaServer 4100 computers. When you choose this option, the console tests a minimum amount of memory and leaves the rest for the operating system to test.

To use this new feature, you need to specify a value for the MEMORY_TEST environment variable at the console before booting. The values for MEMORY_TEST are as follows:

Value	Description
FULL (off)	The console does all the testing.
NONE	32 MB of memory are tested before booting.
PARTIAL	256 MB of memory are tested before booting.

If you set MEMORY_TEST to NONE or PARTIAL, OpenVMS tests any remaining untested memory on an as-needed basis at either or both of the following times:

- While the operating system is booting
- In the scheduler idle loop when no processes are available to run

When you change the value of MEMORY_TEST, you must issue the INIT console command before the new value takes effect. Therefore, you need to follow these steps from the console before booting:

1. Change the value of MEMORY_TEST (if desired).
2. Issue the INIT command from the console.
3. Boot the operating system.

OpenVMS also gives you more control over when memory is actually tested. Bit 2 in the system parameter MMG_CTLFLAGS controls deferred memory testing:

- If the bit is clear (the default), OpenVMS tests memory in the background and not necessarily before the bootstrap process has completed.
- If you set the bit, OpenVMS guarantees that all memory will be tested by the end of EXEC_INIT in the system bootstrap process; that is, before IPL is lowered from 31.

3.5 DIGITAL Cluster MIB Subagent Implemented on OpenVMS

The DCM (Digital Cluster MIB) is a private Digital MIB (Management Information Base) that delivers management information about OpenVMS Cluster systems.

The Extensible Simple Network Management Protocol (eSNMP) makes it possible for network managers to manage many different types of devices across all network and vendor boundaries through the use of databases called MIBs (Management Information Bases). Essentially, information is exchanged between **master agents** and **subagents**, which are devices such as routers and servers on the network being managed, and **managers**, which are the devices on the network through which the management is done.

The DCM consists of two extensions, or subagents:

- System—Describes a management interface to cluster system information not defined by standard MIBs.
- Management—Describes instrumentation in the DIGITAL extension agent, including the ability to detect and monitor thresholds on integer variables.

The representation of the DCM within the standard Structure of Managed Information (SMI) framework is:

```
iso(1) org(3) dod(6) internet(1) private(4) enterprises(1) 36
```

OpenVMS Version 7.2 implements the DCM subagents. With the DCM subagents, you can remotely determine status information for an OpenVMS Cluster system, including the following:

- Cluster software version
- Status of the cluster software: installed, running, failed, and so forth
- Type of cluster that is running
- State change to reflect when a cluster member is added or deleted

To access the DCM subagents, you use the following software:

- DIGITAL's ServerWORKS Manager Version 3.0 or any MIB browser that has access to the DCM definitions.
- DIGITAL TCP/IP Services for OpenVMS Version 4.1 or later. The DCM subagents use the SNMP agent supplied with TCP/IP Services to communicate with SNMP clients.

The following sections describe the DCM subagents and explain how to set up your system to use them.

3.5.1 Overview of DCM Subagents

DCM subagents respond to SNMP requests for a DCM **object**—the data item that the network manager is concerned with, or a **trap**—information about a change of status. A subagent is responsible for reporting on and maintaining the data pertaining to these objects and traps.

The DCM subagents implement the objects listed in Table 3–1. Each object returns information relevant to an OpenVMS Cluster system and can be accessed by a network manager through ServerWORKS Manager.

Table 3–1 DCM Subagent Objects Implemented on OpenVMS

Object	Data Type	Access	Description
Cluster Information			
svrCluSoftwareVendor	DisplayString	Read only	Cluster software vendor name. The current value is Digital.
svrCluSoftwareVersion	DisplayString	Read only	Cluster software version. This is the OpenVMS version string.
svrCluSoftwareStatus	ClusterStatus	Read only	The status of the cluster software. Possible values are running and not running.
svrCluClusterType	ClusterType	Read only	The type of cluster that is running. The current value is OpenVMS.
svrCluExtensionOID	Object Identifier	Read only	The authoritative identification for the MIB. If no such identifier exists, the value {0.0} is returned.
svrCluThisMember	Integer	Read only	Index into the member table (svrCluMemberTable) that corresponds to this node.
SMNP Traps			
svrCluMemberAdded	Trap Packet	Read only	Generated when a cluster member is added.
svrCluMemberDeleted	Trap Packet	Read only	Generated when a cluster member is deleted.

(continued on next page)

Table 3–1 (Cont.) DCM Subagent Objects Implemented on OpenVMS

Object	Data Type	Access	Description
Node-Specific Information			
svrCluMemberIndex	Integer	Read only	A unique index for the entry. Values of svrCluMemberIndex must remain constant at least between reboots of the network management system on the managed node. Where possible, this value should reflect the system's native member identifier.
svrCluMemberName	DisplayString	Read only	The SCS node name of this cluster member. A zero-length value means the member's node name is unknown. This name may not necessarily resolve to an address.
svrCluMemberComment	DisplayString	Read only	This is the hardware name of the node, as returned by the $GETSYI system service.
svrCluMemberStatus	MemberStatus	Read only	Status of this member. Possible values are normal and removed.
svrCluMemberAddressIndex	Integer	Read only	The index for this address.
svrCluMemberAddress	IpAddress	Read only	An IP address of this cluster member. This address may not be reachable from nodes that are not configured into the cluster.

3.5.2 Setting Up the System to Use the DCM Subagents

For your convenience, the following files have example entries to start, run, and shut down the DCM subagents.

- [TCPIP$SNMP]TCPIP$EXTENSION_MIB_STARTUP.COM. (This file is called by TCPIP$SNMP_STARTUP.COM.)
- [TCPIP$SNMP]TCPIP$EXTENSION_MIB_RUN.COM. (This file is called by TCPIP$SNMP_RUN.COM.)
- [TCPIP$SNMP]TCPIP$EXTENSION_MIB_SHUTDOWN.COM. (This file is called by TCPIP$SNMP_SHUTDOWN.COM.)

Search the files for SVRCLUSTER_MIB entries and edit them as needed. You must also remove the `GOTO` entries, which cause the command procedures to simply exit.

3.6 Ethernet and Fast Ethernet Support (Alpha Only)

OpenVMS Alpha Version 7.2 provides run-time and boot support for the DE500-BA PCI Ethernet network interface card, which is capable of 10 or 100 Mb/s operation in half or full-duplex mode. OpenVMS Alpha Version 7.2 also provides run-time and boot support for the DE500-FA PCI Fast Ethernet network interface card, which is capable of 100 Mb/s operation in half or full-duplex mode.

For more information about these adapters, see Section 4.1.

3.7 Extended File Specifications (Alpha Only)

Extended File Specifications is a feature of OpenVMS Alpha Version 7.2 that allows the use of Windows-style file specifications. Extended File Specifications includes support for the following:

- The ODS-5 disk structure. ODS-5 is an extension to the existing ODS-2 disk structure, and adds the ability to use extended file names that can be more easily mapped between Windows and OpenVMS. ODS-5 expands the available character set and filename length to be consistent with Windows 95 and Windows NT, and sets the stage for future Unicode file name support through PATHWORKS.
- Deeper directories. Enhancements to RMS provide deep directory support on both ODS-2 and ODS-5 volumes.

Taken together, these components provide much greater flexibility for OpenVMS Alpha systems (using Advanced Server for OpenVMS 7.2, formerly known as PATHWORKS for OpenVMS), to store, manage, serve, and access files that have names similar to those in a Windows 95 or Windows NT environment.

This section provides an overview of the benefits, features, and support for Extended File Specifications, as well as changes in OpenVMS behavior that occur when using Extended File Specifications. Refer to *OpenVMS Guide to Extended File Specifications* for additional information on Extended File Specifications. Online help on Extended File Specifications is available by typing the following:

```
$ HELP Ext_File_Specs
```

3.7.1 Benefits of Extended File Specifications

The deep directories and extended file names supported by Extended File Specifications provide the following benefits:

- OpenVMS system managers can manage files with extended names and deep directories in the same manner as Windows NT users.
- Users of Advanced Server for OpenVMS 7.2 (formerly known as PATHWORKS for OpenVMS) have the ability to store longer file names and use deeper directory structures, which are more compatible with Windows 95 and Windows NT file names.
- Applications developers who are porting applications from other environments that have support for deep directories can use a parallel structure on OpenVMS.
- Longer file naming capabilities and Unicode support enables OpenVMS Version 7.2 to act as a DCOM server for Windows NT clients, and ODS-5 provides capabilities that make the OpenVMS and Windows NT environment more homogeneous for DCOM developers.
- Java applications on OpenVMS will comply with Java object naming standards.
- General OpenVMS users can make use of long file names, new character support, and the ability to have lowercase and mixed-case file names.

These benefits result from the features described in Section 3.7.2.

3.7.2 Features of Extended File Specifications

Extended File Specifications consists of two main features, the ODS-5 volume structure, and support for deep directories.

3.7.2.1 ODS-5 Volume Structure

OpenVMS Version 7.2 implements On-Disk Structure Level 5 (ODS-5). This structure provides the basis for creating and storing files with extended file names. You can choose whether or not to enable ODS-5 volumes on your OpenVMS Alpha systems. For more information about the structure of an ODS-5 disk, refer to the *OpenVMS Guide to Extended File Specifications*.

The ODS-5 volume structure allows the following features:

- Long file names
- More characters legal within file names
- Preservation of case within file names

3.7.2.2 Deep Directory Structures

Both ODS-2 and ODS-5 volume structures support deep nesting of directories, subject to the following limits:

- There can be up to 255 levels of directories.
- The name of each directory can be up to 236 8-bit or 117 16-bit characters long.

Complete file specifications longer than 255 bytes are abbreviated by RMS when presented to unmodified applications.

For example, a user can create the following deeply nested directory:

```
$ CREATE/DIRECTORY [.a.b.c.d.e.f.g.h.i.j.k.l.m]
```

A user can create the following directory with a long name on an ODS-5 volume:

```
$ CREATE/DIRECTORY
[.AVeryLongDirectoryNameWhichHasNothingToDoWithAnythingInParticular]
```

3.7.2.2.1 Directory Naming Syntax On an ODS-5 volume, directory names conform to most of the same conventions as file names when using the ISO Latin-1 character set. Periods and special characters can be present in the directory name, but in some cases, they must be preceded by a circumflex (^) in order to be recognized as literal characters.

3.8 Fast Skip for SCSI Tape Drives

There is a new qualifier for the SET MAGTAPE command, /FAST_SKIP, that allows you to skip by file mark or by record. The options are PER_IO (default), ALWAYS, and NEVER.

Note

This tape positioning qualifier is for use on local SCSI tape drives only.

To find out if your drive supports the fast skip option, use the SHOW DEVICE command.

For more information, refer to the *OpenVMS DCL Dictionary: N–Z* and online help.

3.9 FastTrack Web Server (Alpha Only)

Netscape FastTrack, a robust Web server for creating, building, publishing, and serving Web pages and applications, is now bundled at no additional cost with OpenVMS Alpha Version 7.2. FastTrack's rich cross-platform environment supports building or deploying a variety of Web or database applications. Based on open Internet and intranet standards, FastTrack enables high-fidelity communication and information sharing within and beyond the organization. FastTrack Server offers a rich environment for building or deploying many types of web or database applications, while remaining extremely easy to use and manage.

FastTrack takes the cost and complexity out of building a Web site. Users can install FastTrack easily, get it up and running, and publish to it in a matter of minutes. With FastTrack's Web-based administration interface, users can manage a Web site from any desktop in the organization. With the push of a button, users can publish documents to the server.

FastTrack provides common user and group management across multiple servers by supporting LDAP (Lightweight Directory Access Protocol) for directory access, enabling administrators to specify access rights to documents in a fine-grained manner. FastTrack delivers reliability in its core Web engine through optimized caching, advanced use of kernel threads, HTTP V1.1 support, and sophisticated memory management. FastTrack's security features restrict access to information stored on the server by the following:

- Supporting SSL (secure sockets layer) V3.0, the widely accepted Internet security standard that encrypts the information flow between the server and a Web client
- Supporting client authentication to restrict access to the server based on client certificates

Some of the features that distinguish FastTrack from Netscape Communications and Commerce servers include:

- Web engine based on the HTTP 1.1 standard
- Comprehensive management of users and groups, configurations, and access control
- Client authentication
- Flexible, fine-grained access controls on documents and directories
- Support for the Lightweight Directory Access Protocol (LDAP), so that users and groups can be centrally managed across an organization or even outside it
- Distributed administration that allows administrative privileges to be shared by multiple people or a subset of capabilities to be delegated
- An advanced platform for building crossware applications through its support of the latest versions of Java, JavaScript, LiveConnect, and the web application interface (WAI)
- Centralized certificate management

3.9.1 Benefits

The benefits of using FastTrack for OpenVMS Alpha include:

- Provides a Web-based administration interface, making it easy to manage a site from anywhere in the organization
- Allows users to set up a Web server and publish documents quickly
- Supports industry standards HTTP V1.1 and SSL (secure sockets layer) V3 for encrypting information flow between the client and server
- Provides a web-based administration interface, making it easy to manage a site from anywhere in the organization
- Includes LDAP support for centralized user and group management
- Enables building and deployment of crossware applications with server-side Java and JavaScript

3.9.2 Availability and Ordering

Netscape FastTrack Server for OpenVMS Alpha is included in the OpenVMS Alpha Version 7.2 release and is licensed under the OpenVMS license, QL-MT1*. No additional license is required. Netscape FastTrack Version 3.03 for OpenVMS Alpha software, which runs on OpenVMS Alpha Version 7.1 or later, is also available at no cost as a downloadable kit from the OpenVMS web site at:

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

3.10 Fibre Channel Support (Alpha Only)

Fibre Channel is a new ANSI standard network and storage interconnect that offers many advantages over other interconnects. Its most important features are described in Table 3–2.

Note

To help you plan for the introduction of Fibre Channel in your computing environment, this documentation is provided in advance of the Fibre Channel functionality, which will be available shortly after the release of OpenVMS Version 7.2.

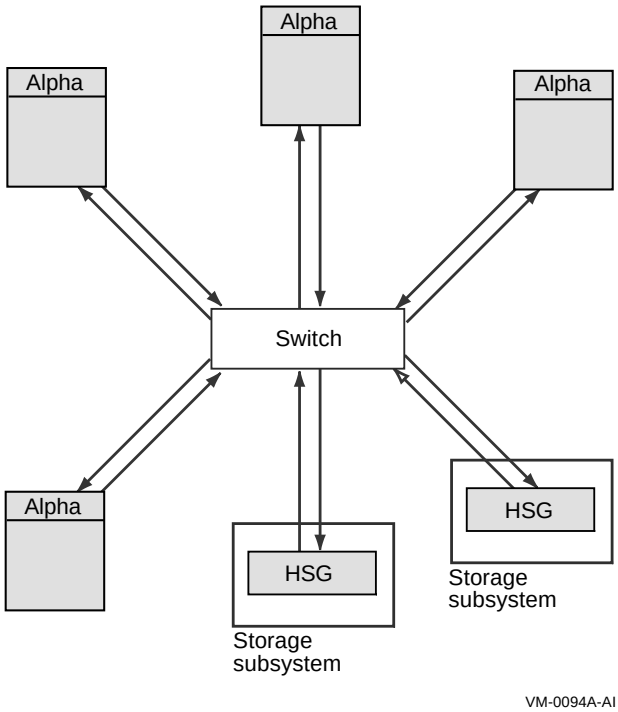
Table 3–2 Fibre Channel Features

Feature	Description
High-speed transmission	1.06 gigabits per second, full duplex, serial interconnect (can simultaneously transmit and receive 100 megabytes of data per second)
Choice of media	Initial OpenVMS support for fibre-optic media. Potential future support for copper media.
Long interconnect distances	Initial OpenVMS support for multi-mode fiber at 400 meters per link. Potential future support for 10 kilometer single-mode fiber links and 30 meter copper links.
Numerous topologies	Initial OpenVMS support for switched FC (highly scalable, multiple concurrent communications). Potential future support for arbitrated loop (maximum number of nodes fixed at 126, shared bandwidth, hardware relatively inexpensive), fabric of switches, mixed arbitrated loop and switches, and point-to-point

The initial OpenVMS implementation supports a single-switch topology, with multi-mode fiber-optic media, at distances up to 400 meters per link.

Figure 3–1 shows a logical view of a switched topology. The FC nodes are either Alpha hosts, or storage subsystems.

Figure 3–1 Switched Topology, Logical View



Each link from a node to the switch is a dedicated FC connection. The switch provides store-and-forward packet delivery between pairs of nodes. Concurrent communication between disjoint pairs of nodes is supported by the switch.

For more information about OpenVMS support for Fibre Channel, refer to *Guidelines for OpenVMS Cluster Configurations*.

3.11 Firmware Revision Checking (Alpha Only)

OpenVMS Alpha Version 7.2 provides enhanced firmware checking for systems during a boot operation. When you boot the OpenVMS Alpha operating system CD–ROM, the system automatically checks the version of console firmware (including PALcode) that is running on your computer. The system also provides more explicit information about how and when to upgrade the firmware. Previously, only the PALcode version was checked and the informational messages were less specific.

Note that once you install this version of the OpenVMS Alpha operating system, the firmware check (similar to the previous PALcode check) will occur each time you reboot the system.

3.12 InfoServer Service Listing: Improved Display and Service Selection

OpenVMS Version 7.2 offers system managers a quicker way to view and select services from a Compaq InfoServer. The service listings that are displayed during both a MOP boot and at run time (with the MCR ESS$LADCP SHOW SERVICES command) have been reorganized.

Prior to Version 7.2, service information was displayed in random order, such as shown in the following excerpt from an OpenVMS Version 7.1 MOP boot procedure:

```
   .
   .
   .
Servers found:  19

 Service Name Format:
        Service Number
        Service Name
        Server Name
        Ethernet ID

#1
X6FA_IMAGE
VMS_DSS
08-00-2B-27-FC-0F

#2
X6G7_KIT
ALPHA_DSS
08-00-2B-28-17-A3

#3
CDDOC06JUL21
SUPER
08-00-2B-20-CD-07

 Enter a Service Number or <CR> for more:
   .
   .
   .
```

With a maximum of four services per screen, this selection process often involved repeated scrolling to locate the correct services.

Now service information is grouped by server and displayed alphabetically, as shown by the following example taken from an OpenVMS Version 7.2 MOP boot procedure:

```
   .
   .
   .
 Servers found::   19

 Services offered by node AJAX_TEST_SERVR, Address: 08-00-2B-2D-B8-32

#   2  VAXDOCSEP931
#   3  R6LBCK
#   4  V71R01-03
#   5  CDDOC09NOV22

 Services offered by node IMS$DSS1, Address: 08-00-2B-23-94-15
#   6  VXT_UPDATE

 Services offered by node PUB_08002EE0AF60, Address: 08-00-2B-B0-AF-60

#   1  FREEWAREV20

 Services offered by node VMS$DSS1, Address: 08-00-2B-2C-D4-A6
#   7  VAXBINDEC969
#   8  CDBIN_01
#   9  CDBIN_02
   .
   .
   .
```

The run-time service information is also grouped by server, as follows:

```
$ MC ESS$LADCP SHO SER

Interrogating network for Disk services, please wait...

Disk services offered by node AXX_150 (LAD V3.1, Address: 08-00-2B-26-A4-B5)

                                          Current   Writes
Service:                    Device:   Rating: Connects: Allowed?:

PCSI_MOTIF123                RZ57      65263       0     Yes
VPWRK_LATEST                 RZ57      65272       0     Yes
VPWRKV5D                     RZ57      65272       0     Yes
APWRK_LATEST                 RZ57      65263       0     Yes
VAXVMS061                    RZ57      65263       0     Yes
VMS060                       RZ57      65272       0     Yes
VMS054                       RZ57      65272       0     Yes
Disk services offered by node FPG90 (LAD V3.1, Address: AA-00-04-00-F3-FC)

                                          Current   Writes
Service:                 Device:      Rating: Connects: Allowed?:
RSM$FETCH_CSG20-VMS       OpenVMS AXP     64       0     No
RSM$FETCH_CSG20-AVMS      OpenVMS AXP     64       0     No
```

This improved display format provides more service information per screen and enables system managers to monitor and select services more easily.

3.13 LANCP Define Device and Set Device for Classical IP: New Qualifier (Alpha Only)

OpenVMS Version 7.2 has two new qualifiers, /PVC and /NOPVC, for Define Device and Set Device commands for LANCP. Table 3–3 shows the qualifiers for the Define Device and Set Device commands.

Table 3–3 Define Device and Set Device

Qualifier	Description
/PVC=([vc-id],...)	On Alpha systems, defines or sets the permanent virtual channel (PVC). This is an optional qualifier. An example of enabling the PVC is: `/PVC = (200,105)`
/NOPVC=([vc-id],...)	On Alpha systems, does not define or set the permanent virtual channel (PVC). This is an optional qualifier.

3.14 LANCP Classical IP (CLIP) Qualifier: New and Changed Keyword (Alpha Only)

OpenVMS Version 7.2 has a new and changed keyword for the /CLIP qualifier for the Define Device and Set Device LANCP commands. The description for the keyword type=server has changed and the new keyword is shown in Table 3–4.

Table 3–4 Keyword for /CLIP Qualifier for Define Device and Set Device

Qualifier	Description
/CLIP	The meanings for the syntax of keyword and subkeyword for /CLIP are as follows: • *type=server* Starts up a CLIP server. Only one server for each LIS is allowed, and the server needs to be started first. • *type=(server,client)* Starts up a CLIP server and client.

3.15 Monitor Utility: TCP/IP Support Added

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

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

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

See SYS$STARTUP:SYSTARTUP_VMS.TEMPLATE for examples.

3.16 OpenVMS Cluster Systems

The following new OpenVMS Cluster features are described in this section:

- New CIPCA adapter support
- Clusterwide logical names
- Intra-cluster communication (ICC) system services
- Lock Manager enhancements
- MEMORY CHANNEL enhancements

- Multipath SCSI configurations with parallel SCSI or Fibre Channel
- SCSI OpenVMS Cluster system supports four nodes
- Ultra SCSI configuration support
- Warranted and migration support

3.16.1 New CIPCA Adapter Support

The CIPCA adapter is a PCI-to-CI storage host bus adapter. CIPCA enables users to connect their existing CI-based storage to high-performance PCI-based AlphaServer systems.

Since the release of OpenVMS Version 7.1, the CIPCA-BA adapter has been qualified on OpenVMS Version 7.1. The CIPCA-BA adapter requires two PCI slots. The earlier CIPCA-AA adapter requires one EISA slot and one PCI slot. The CIPCA-BA is intended for newer servers that have a limited number of EISA slots.

The CIPCA adapter enables users to include the latest Alpha-based servers within an existing OpenVMS Cluster system, thereby taking advantage of servers offering the best price performance while maintaining previous investments in storage subsystems. The CIPCA adapter and the HSJ50 storage controller allow PCI to CI connectivity and increase the performance of the CI with support for 4K packets.

The maximum number of CIPCA adapters that can be configured within a system is dependent upon the system, the available number of EISA and PCI slots, the combination of CIPCA models selected, as well as other system options. For more information about CIPCA, see *Guidelines for OpenVMS Cluster Configurations*.

3.16.2 Clusterwide Logical Names

Clusterwide logical names are an extension to the existing logical name support in OpenVMS. They are available on both OpenVMS Alpha and OpenVMS VAX.

This section provides information about clusterwide logical names for system managers. For programming aspects of clusterwide logical names, see Section 4.2.

3.16.2.1 Overview

Clusterwide logical names extend the convenience and ease-of-use features of shareable logical names to OpenVMS Cluster systems. Existing applications can take advantage of clusterwide logical names without any changes to the application code. Only a minor modification to the logical name tables referenced by the application (directly or indirectly) is required.

New logical names created on OpenVMS Version 7.2 are local by default. Clusterwide is an attribute of a logical name table. In order for a new logical name to be clusterwide, it must be created in a clusterwide logical name table.

3.16.2.2 Features

Some of the most important features of clusterwide logical names are:

- When a new node running Version 7.2 joins the cluster, it automatically receives the current set of clusterwide logical names.
- When a clusterwide logical name or name table is created, modified, or deleted, the change is automatically propagated to every other node in the cluster running Version 7.2. Modifications include security profile changes to a clusterwide table.

- Translations are done locally so there is minimal performance degradation for clusterwide name translations.
- Because LNM$CLUSTER_TABLE and LNM$SYSCLUSTER_TABLE exist on all systems running OpenVMS Version 7.2, the programs and command procedures that use clusterwide logical names can be developed, tested, and run on nonclustered systems.

For more information about clusterwide logical names, refer to the *OpenVMS Cluster Systems* manual.

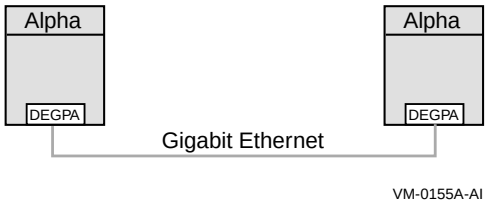
3.16.3 Gigabit Ethernet as a Cluster Interconnect

Note

OpenVMS Cluster support for Gigabit Ethernet will be available shortly after the release of OpenVMS Version 7.2. This documentation is provided to help you plan for the introduction of Gigabit Ethernet in your OpenVMS Cluster configurations,

OpenVMS Alpha Version 7.2 supports Gigabit Ethernet as a cluster interconnect. The nodes in a Gigabit Ethernet OpenVMS Cluster system are connected to a Gigabit Ethernet switch, or, if there are only two nodes, they can be connected point-to-point so that no switch is needed, as shown in Figure 3–2.

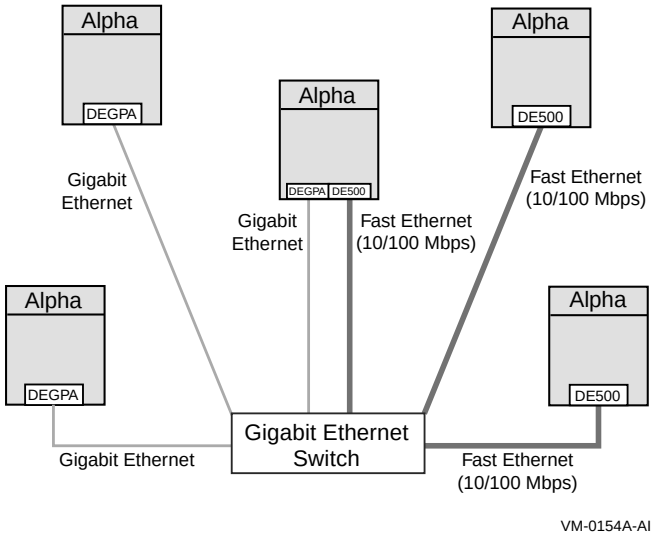
Figure 3–2 Point-to-Point Gigabit Ethernet OpenVMS Cluster Configuration



Most Gigabit Ethernet switches can be configured with Gigabit Ethernet or a combination of Gigabit Ethernet and Fast Ethernet (100 Mbps). Each node can have a single connection to the switch or can be configured with multiple connections. For example, a node can be connected by Gigabit Ethernet and by Fast Ethernet.

Figure 3–3 shows a five-node cluster with two nodes connected by Gigabit Ethernet, one node connected by both Gigabit Ethernet and Fast Ethernet, and two nodes connected by Fast Ethernet. Note that the currently supported DIGITAL PCI-to-Gigabit Ethernet adapter is known as a DEGPA. The currently supported Fast Ethernet family of adapters is named DE50*x*-*xx*.

Figure 3–3 Switched Gigabit Ethernet OpenVMS Cluster Configuration



In a multipath configuration where a node is connected to the switch by two or more cables, such as the middle node shown in Figure 3–3, if one path fails, the remaining path can assume the load of the failed path.

3.16.3.1 System Support

Gigabit Ethernet is supported as a cluster interconnect on several AlphaServer models, as shown in Table 3–5.

Table 3–5 AlphaServer Support for Gigabit Ethernet Adapters

AlphaServer Model	Maximum Number of Adapters	Minimum Memory
AlphaServer GS140	4	128[1]
AlphaServer 8400	4	128[1]
AlphaServer 8200	4	128[1]
AlphaServer 4*x*00	4	128[1]
AlphaServer 1200	2	128[1]
AlphaServer 800	2	128[1]

[1]256 MB is strongly recommended.

3.16.3.2 OpenVMS Cluster Functions Planned for Future Release

A limited number of cluster functions will not be supported in OpenVMS Version 7.1–2 or OpenVMS Version 7.2, as described in Table 3–6. Support for these cluster functions is planned for a future release.

Table 3–6 Cluster Functions Not Currently Supported

Function	Comment
Jumbo frames (frame size >1518 and <9,018)	Jumbo frames are supported at this time over Gigabit Ethernet but not for cluster communications. The frame size supported for cluster communications is the standard 1518-byte maximum Ethernet frame size.
Optimum path selection	Because optimum path selection is not implemented in this release, you cannot rely on the cluster software to always select the optimal path.
Satellite booting with the DEGPA as the boot device	Although the DEGPA cannot be used as the boot device, satellites can be booted over standard 10/100 Ethernet network adapters configured on a Gigabit switch.

3.16.4 Intra-Cluster Communication (ICC) System Services

The new intra-cluster communications (ICC) system services provide an applications programming interface (API) for applications that will run in an OpenVMS Cluster system. Using these services, application program developers can create connections between different processes on the same or different systems within a single OpenVMS Cluster system. For more information about the new ICC system services, refer to the *OpenVMS System Services Reference Manual: GETQUI–Z*, and Section 4.11 in this manual.

3.16.5 Lock Manager Improvements

The lock manager synchronizes resources in an OpenVMS system. For OpenVMS Version 7.2, the lock manager software has been enhanced to improve the performance of applications that issue a large number of lock manager requests and to improve application scaling. These improvements pertain to single systems and to OpenVMS Cluster systems.

For OpenVMS Alpha, the changes include a new location for most of the lock manager data structures. These data structures now reside in S2 space in a structure known as a Pool Zone. The SHOW MEMORY display has been modified to display attributes of the Pool Zone memory used by the lock manager. For more information, refer to the *OpenVMS DCL Dictionary*.

3.16.6 MEMORY CHANNEL Enhancements (Alpha Only)

MEMORY CHANNEL is a high-performance cluster interconnect technology for PCI-based Alpha systems. It is suitable for applications that must move large amounts of data among nodes, such as high-performance databases.

MEMORY CHANNEL supports node-to-node cluster communication only. A second interconnect is required for network traffic or storage access.

When introduced in OpenVMS Alpha Version 7.1, MEMORY CHANNEL supported a maximum of 4 nodes in a 10-foot radial topology. OpenVMS Alpha Version 7.1–1H1 provided the following enhancements for MEMORY CHANNEL Version 1.5:

- Support for a maximum of 8 nodes in a configuration
- Support for a new MEMORY CHANNEL adapter (CCMAA–BA), which provides improved performance
- Time stamps on all messages

- More robust performance in the presence of errors

MEMORY CHANNEL Version 2.0, supported by OpenVMS Alpha Version 7.2, provides the following new capabilities:

- Support for a new adapter (CCMAB–AA) and a new hub (CCMHB–AA)
- Support for simultaneous communication between four sender-receiver pairs
- Support for longer cables for a radial topology up to 3 km
  - Copper cables (3 sizes) up to a 10-m (32.8 ft) topology
  - Fiber-optic cables up to a 3-km topology

  Fiber-optic cables up to 30 meters are available from Compaq. Fiber-optic cables up to 3 kilometers are available from other vendors.

You can configure a computer in an OpenVMS Cluster system with both a MEMORY CHANNEL Version 1.5 hub and a MEMORY CHANNEL Version 2.0 hub. However, the version number of the adapter and the cables must match the hub's version number for MEMORY CHANNEL to function properly.

In other words, you must use MEMORY CHANNEL Version 1.5 adapters with the MEMORY CHANNEL Version 1.5 hub and MEMORY CHANNEL Version 1.5 cables. Similarly, you must use MEMORY CHANNEL Version 2.0 adapters with the MEMORY CHANNEL Version 2.0 hub and MEMORY CHANNEL Version 2.0 cables.

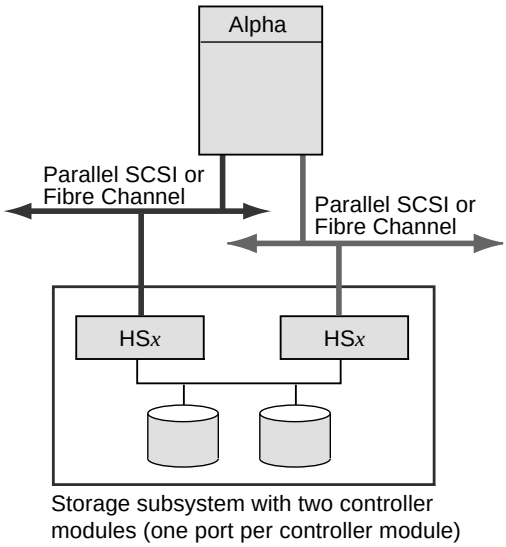
For more information about MEMORY CHANNEL, refer to *Guidelines for OpenVMS Cluster Configurations*.

3.16.7 Multipath SCSI Configurations with Parallel SCSI or Fibre Channel

OpenVMS Alpha Version 7.2 introduces multipath support for parallel SCSI configurations. Shortly after the release of OpenVMS Version 7.2, multipath support will also be available for Fibre Channel configurations.

SCSI multipath support means support for failover between multiple paths that may exist between an OpenVMS system and a SCSI device, as shown in Figure 3–4. If the current path to a mounted disk fails, the system will automatically fail over to the alternate path.

Figure 3–4 Multipath SCSI Configuration



Multipath support is provided for:

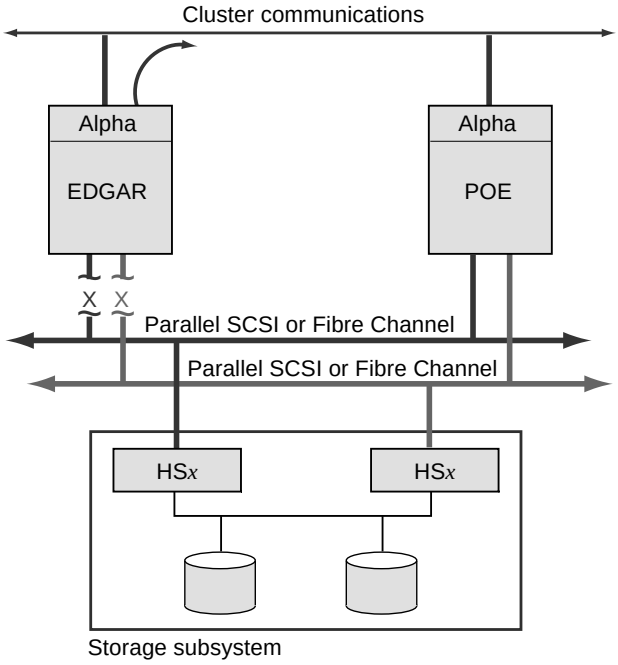
- Systems that are configured with multiple direct connections to the HSZ70, HSZ80, and the HSG80 storage controllers
- Systems that are in a cluster and have one or more direct paths to a disk as well as an MSCP-served path

Note

Multipath support for failover between direct SCSI paths and MSCP served paths will be available soon after the release of OpenVMS Version 7.2.

Figure 3–5 shows a SCSI multipath configuration with multiple direct connections to the HS*x* controllers as well as MSCP served paths to the HS*x* controllers.

Figure 3–5 Direct SCSI and MSCP Served Paths



Multipath SCSI devices can be directly attached to Alpha systems and served to Alpha or VAX systems.

SCSI multipath failover for redundant paths to a storage device greatly improves data availability and, in some configurations, improves performance.

For more information about SCSI multipath support, refer to *Guidelines for OpenVMS Cluster Configurations*.

3.16.8 SCSI OpenVMS Cluster System Supports Four Nodes

With the introduction of the SCSI hub DWZZH-05, four nodes can now be supported in a SCSI multihost OpenVMS Cluster system. In order to support four nodes, the hub's fair arbitration feature must be enabled. This hub is supported with either KZPSA or KZPBA-CB adapters.

This configuration is supported on the following versions of OpenVMS Alpha:

- Version 7.1–1H1
- Version 7.1–1H2
- Version 7.1–2
- Version 7.2

Prior to the introduction of the SCSI hub DWZZH-05, a maximum of three nodes were supported in a SCSI multihost OpenVMS cluster system.

3.16.9 Ultra SCSI Configuration Support

OpenVMS Alpha Version 7.1–1H1 introduced support for certain Ultra SCSI devices in Ultra SCSI mode in single-host configurations. Since the release of OpenVMS Alpha Version 7.1–1H1, additional Ultra SCSI devices have been qualified on OpenVMS, and support for multihost configurations has been added to OpenVMS.

OpenVMS Version 7.2 includes Ultra SCSI support for both single-host and multihost configurations. A maximum of four hosts is supported in an Ultra SCSI multihost configuration when a DWZZH-05 hub is used with fair arbitration enabled.

Table 3–7 summarizes the Ultra SCSI support provided by OpenVMS, including support for several significant Ultra SCSI devices. For information about all Ultra SCSI devices supported by OpenVMS and about configuring OpenVMS Alpha Ultra SCSI clusters, see the documents described in Table 3–9.

Table 3–7 OpenVMS Alpha Ultra SCSI Support

Configuration/Adapter	Version	Description
Single-host configurations using the KZPBA-CA	7.1–1H1	The KZPBA-CA is a single-ended adapter. The KZPAC Ultra SCSI host RAID controller is also supported in single-host configurations.
Single-host configurations using the KZPBA-CB	7.1–1H1	The KZPBA-CB is a differential adapter. The HSZ70 is also supported in Ultra SCSI mode, using the KZPBA-CB.
Multihost configurations using the KZPBA-CB	7.1–1H1 with a remedial kit	Up to four hosts can share the Ultra SCSI bus when a DWZZH-05 hub is used with fair arbitration enabled. The HSZ70 is also supported on the multihost bus.
Multihost configurations using the KZPBA-CB	7.2	Up to four hosts can share the Ultra SCSI bus when a DWZZH-05 hub is used with fair arbitration enabled. The HSZ70 is also supported on the multihost bus.

Note the restrictions described in Table 3–8.

Table 3–8 OpenVMS Restrictions

Restriction	Comments
Firmware for the KZPBA–CB must be Version 5.53 or higher.	Earlier firmware versions do not provide multihost support.
Console firmware must be updated with the Alpha Systems Firmware Update CD Version 5.1 or higher.	All console SCSI driver fixes are included on this CD. This CD also includes the latest version of the KZPBA-CB firmware (Version 5.53 or higher).

Table 3–9 provides pointers to additional documentation for Ultra SCSI devices and for configuring OpenVMS Alpha Ultra SCSI clusters.

Table 3–9 Documentation for Configuring OpenVMS Alpha Ultra SCSI Clusters

Topic	Document	Order Number
SCSI devices that support Ultra SCSI operations and how to configure them	*StorageWorks UltraSCSI Configuration Guidelines*	EK–ULTRA–CG
KZPBA–CB UltraSCSI storage adapter	*KZPBA–CB UltraSCSI Storage Adapter Module Release Notes*	AA–R5XWA–TE
Multihost SCSI bus operation in OpenVMS Cluster systems	*Guidelines for OpenVMS Cluster Configurations*	AA–Q28LB–TK
Systems and devices supported by OpenVMS Version 7.1–1H1	*OpenVMS Operating System for Alpha and VAX, Version 7.1–1H1 Software Product Description*	SPD 25.01.*xx*
Multihost SCSI support	*OpenVMS Cluster Software Software Product Description*	SPD 29.78.*xx*

Information about StorageWorks Ultra SCSI products is available and periodically updated on the World Wide Web at the following URL:

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

OpenVMS software product descriptions are also available and periodically updated on the World Wide Web at the following URL:

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

You will find the software product descriptions under Publications, a choice on the home page.

3.16.10 Warranted and Migration Support

OpenVMS Alpha Version 7.2 and OpenVMS VAX Version 7.2 provide 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.

Migration support helps customers move to warranted OpenVMS Cluster version mixes with minimal impact on their cluster environments.

Table 3–10 shows the level of support provided for all possible version pairings.

Table 3–10 OpenVMS Cluster Warranted and Migration Support

	Alpha V6.2–*xxx*	Alpha V7.1–*xxx*	Alpha V7.2
VAX V6.2–*xxx*	WARRANTED	Migration	Migration
VAX V7.1–*xxx*	Migration	WARRANTED	Migration
VAX V7.2	Migration	Migration	WARRANTED

For OpenVMS Version 6.2 nodes to participate in a cluster with systems running either Version 7.1 or Version 7.2, the cluster compatibility kit must be installed on each Version 6.2 node. In addition, if you use the Monitor Utility in a mixed version utility, you must install a new remedial kit. For more information about these kits, refer to the *OpenVMS Version 7.2 Release Notes*.

Compaq does not support the use of Version 7.2 with Version 6.1 (or earlier versions) in an OpenVMS Cluster. In many cases, mixing Version 7.2 with versions prior to Version 6.2 will successfully operate, but Compaq cannot commit to resolving problems experienced with such configurations.

Note

Nodes running OpenVMS VAX Version 5.5–2 or earlier versions, or OpenVMS Alpha Version 1.0 or 1.5, cannot participate in a cluster with one or more OpenVMS Version 7.2 nodes. For more information, refer to the *OpenVMS Version 7.2 Release Notes*.

3.17 OpenVMS Management Station Version 3.0

Version 3.0 of OpenVMS Management Station adds storage management support to the existing user account and printer management features.

OpenVMS Management Station now makes it easy for you to manage your disk storage devices across multiple OpenVMS Cluster systems and OpenVMS nodes. With Version 3.0 of OpenVMS Management Station, you no longer need to maintain complicated command procedures to control your storage environment. You can create, delete, and manage storage from an easy-to-use Windows interface.

OpenVMS Management Station provides a persistent database that can automatically determine and configure your system's storage configuration at system startup.

OpenVMS Management Station is now based on the Microsoft Management Console (MMC). The Microsoft Management Console provides a common management framework for various administration programs. OpenVMS Management Station is implemented as an MMC snap-in and includes all of the MMC components you need.

3.17.1 OpenVMS Management Station New Features

OpenVMS Management Station Version 3.0 includes the following new features:

- Storage Configuration Management. You can create, delete, and manage a disk volume under one interface. Wizards make it easy to create a volume.
- Configuration Control Over Reboots. If you allow it to, OpenVMS Management Station preserves the disk configuration across reboots. OpenVMS Management Station can mount and maintain your storage configuration without intervention.

  And, OpenVMS Management Station mounts volumes much faster than is possible with DCL or command procedures, so performance is enhanced.
- Coexistence with Established Environment. You do not have to change your existing DCL command procedures if you do not want to. You can use the display capabilities of OpenVMS Management Station without having to use the automated mount feature. In this way, you can give OpenVMS Management Station as little or as much control as you are comfortable with.
- Remote Management Supported. You can use OpenVMS Management Station to remotely manage your OpenVMS systems. Once you establish a TCP/IP dialup connection, you can then use OpenVMS Management Station to manage your OpenVMS systems from home, while traveling, and so forth.
- Integrated Web Features. MMC allows you to insert World Wide Web URLs directly into the viewer and access them via Microsoft Internet Explorer. OpenVMS Management Station includes hot links to the OpenVMS and OpenVMS Management Station web pages, where you can find the latest product information and kits. You can use this feature to add web links of your own. For example, you might add a link to an internal Help Desk troubleshooting page, or to a system management procedures page.

3.18 OpenVMS Registry

To allow OpenVMS and Windows NT to interoperate, Compaq has provided a registry on OpenVMS. Both PATHWORKS and COM for OpenVMS use the OpenVMS Registry. Like the Windows NT registry, the OpenVMS Registry is made up of two components: the OpenVMS Registry database and the OpenVMS Registry server. The OpenVMS Registry database is a systemwide or clusterwide hierarchical database of configuration information. This information is stored in a database structure of keys and associated values. The OpenVMS Registry server controls all OpenVMS Registry operations, such as creating and backing up the OpenVMS Registry database, and creating, displaying, modifying, or deleting keys and values.

The OpenVMS Registry includes interfaces (COM APIs and system services) to allow applications to control the OpenVMS Registry server and to read and write to the OpenVMS Registry database. The OpenVMS Registry also includes server management utilities to allow system managers to display and update OpenVMS Registry information from the OpenVMS DCL command line.

The OpenVMS Registry is compatible with the Windows NT registry. Windows NT client applications such as `RegEdt32` can connect to and edit the OpenVMS Registry.

3.18.1 For More Information

For more information about the OpenVMS Registry, see the *OpenVMS Connectivity Developer Guide*. You can find this document in the following locations:

- The document is included in the COM for OpenVMS kit. When you install the kit, the document is placed in the COM for OpenVMS [SYSHLP] subdirectory.
- The document is available from the OpenVMS website at the following location:

  http://www.openvms.digital.com/openvms/products/dcom/index.html

For more information about installing and managing the OpenVMS Registry, refer to the *OpenVMS System Manager's Manual*.

3.19 PCI Gigabit Ethernet Support (Alpha Only)

Run-time support for the DIGITAL PCI-to-Gigabit Ethernet adapter (DEGPA) is included in OpenVMS Alpha Version 7.2. The DEGPA is supported on single systems. For information about OpenVMS Cluster support, see Section 3.16.3.

The DEGPA conforms to the IEEE 802.3z Gigabit Ethernet standard running over 1000BASE-SX fiber optic cabling. It can run point-to-point to another Gigabit Ethernet adapter or to a Gigabit Ethernet hub or switch.

To configure the DEGPA, do the following:

1. Verify that your system is running OpenVMS Alpha Version 7.2.
2. Power down your system.
3. Install the DEGPA in an available PCI slot, preferably a 64-bit slot. (For installation instructions for this adapter, see the *DIGITAL PCI-to-Gigabit Ethernet Adapter Installation and Configuration* guide that accompanies it.)
4. Boot the system.
5. Determine which EW device is the DEGPA with the following command:

   ```
   $ RUN SYS$SYSTEM:LANCP
   LANCP> SHOW DEV EW/CHAR
   ```

   The device that has a line speed of 1000 megabits per second is the DEGPA.
6. Verify that the device is cabled correctly to the switch or point-to-point with another device. If it is, a "link up" console message is displayed. If you missed the console message (it is not logged to the OPCOM log file), you can either check the DEGPA or use the SDA utility:

   - On the DEGPA, there are two small lights. The light next to the link symbol (two circles linked together) is green if the DEGPA is correctly cabled. (For more information about the lights, see the *DIGITAL PCI-to-Gigabit Ethernet Adapter Installation and Configuration* guide.)
   - Issue the SDA command SHOW LAN, as follows:

     ```
     $ ANALYZE/SYSTEM
     SDA> SHOW LAN
     ```

     If cabled properly, SDA displays "ring available" for the device.

3.19.1 Firmware Updates

As firmware is updated for the Gigabit Ethernet adapter, it will be included in successive versions of the driver. Unlike other adapters, the firmware image for the Gigabit Ethernet adapter is embedded in the driver. (The firmware is loaded into the adapter at device initialization and is part of the driver image.) This means that you do not update the firmware.

3.19.2 Link Autonegotiation Support

The OpenVMS Gigabit Ethernet driver supports link autonegotiation. Autonegotiation is enabled by default and can be turned on or off using the SYSMAN utility or the LANCP utility. If you use the SYSMAN utility to change the setting, it takes effect on all the Gigabit Ethernet adapters. The LANCP utility can be used to change the setting on individual adapters or on all of them. The change takes effect immediately when issued by either utility.

Many Gigabit Ethernet switches support link autonegotiation but some, such as the Gigaswitch/Ethernet, do not. If the Gigabit Ethernet switch you are using does not support link autonegotiation, you must disable link autonegotiation on the driver.

Each time the setting of link negotiation is changed, a console message is displayed. At driver initialization, if you have disabled autonegotiation, the following message is displayed on the console:

```
%EWA0, Autonegotiation disabled per SYSGEN parameter LAN_FLAGS
```

Because the Gigabit Ethernet driver checks the LAN_FLAGS setting once every second and changes the link setting as dictated by bit 5, the driver will be able to see a valid link within a few seconds.

If the device is cabled properly but the driver is not able to see a valid link, it is probably because the switch does not support autonegotiation.

Directions for using SYSMAN or LANCP to turn autonegotiation off or on follow.

3.19.2.1 SYSMAN Utility Directions

Using SYSMAN, you can disable or enable autonegotiation on all adapters. You cannot single out individual adapters. To disable autonegotiation, you set bit 5 of the LAN_FLAGS system parameter. To enable autonegotiation, you clear the bit.

To disable autonegotiation:

1. Verify that the device is cabled properly by looking at the link light or by using SDA, as previously described.
2. Invoke SYSMAN and issue the following SYSMAN commands:

   ```
   $ run sys$system:sysman
   SYSMAN> parameters use current
   SYSMAN> parameters set lan_flags %x20
   SYSMAN> parameters write active
   SYSMAN> parameters write current
   SYSMAN> exit
   ```

   The notation "%x" instructs the operating system that the following number is hexadecimal.

Note

The setting for bit 5 should be OR'd with any other existing settings for the LAN_FLAGS system parameter.

3. Insert or edit the following line in SYS$SYSTEM:MODPARAMS.DAT:

```
lan_flags = %x20  ! turn off auto-negotiate on DEGPA
```

To enable autonegotiation, follow the previous directions except clear bit 5 instead of setting it. Then modify the relevant line in SYS$SYSTEM:MODPARAMS.DAT.

3.19.2.2 LANCP Utility Directions

You can use the LANCP utility to disable or enable autonegotiation on one or more specific adapters with separate commands or on all the adapters with a single command.

Issue the following command to disable link autonegotiation for a specific adapter, for example, ewa:

```
$ mc lancp set dev devname/noauto ewa
```

Issue the following command to disable link autonegotiation on all Gigabit Ethernet adapters connected to the Gigabit Ethernet switch:

```
$ mc lancp set dev devname/noauto ew/all
```

If you have disabled link autonegotiation, you can turn it back on by using the syntax of the previous commands but replacing `noauto` with `auto`.

3.19.3 Jumbo Frame Support

The OpenVMS Gigabit Ethernet driver supports frame sizes larger than 1518 bytes, the maximum frame size supported on Ethernet and Fast Ethernet. The Gigabit Ethernet adapter supports frame sizes up to 9018 bytes.

The Gigabit Ethernet driver limits the actual maximum frame size to the size of the non-paged pool lookaside lists less the transmit and receive request frame overhead of 640 bytes. Table 3–11 shows the default frame size and the maximum size allowed for this release and for OpenVMS Version 7.2.

Table 3–11 OpenVMS Frame Size Support

Version	Default Size	Maximum Size
OpenVMS Version 7.1–2	1512 bytes	4480 bytes
OpenVMS Version 7–2	1512 bytes	7552 bytes

Some Gigabit Ethernet switches, such as the Gigaswitch/Ethernet, do not support frame sizes larger than 1518 bytes. Other switches may support the larger frame size, but may require applications to determine whether the larger frame size can be used when communicating with another node on the same switch or elsewhere on the LAN. For example, the switch may accept large frames but may discard them, without notice to the sender, when the destination is a Fast Ethernet node on the same switch.

Each application is responsible for negotiating a frame size between nodes. Some applications, such as Digital TCP/IP, rely on the switch for frame fragmentation or notification of frame size errors (ICMP frame for TCP/IP).

If the switch does not provide jumbo frame support, the maximum frame size must be limited by the LAN driver to 1518 bytes. This limit enables communication over the switch to nodes that do not support the larger frame size or that are limited by switch hardware between two nodes.

Note

Enable the use of jumbo frames if the Gigabit Ethernet switch supports jumbo frames or if you are configured point-to-point with another node that also supports jumbo frames, and if the application also supports jumbo frames.

3.19.3.1 Enabling Use of Frames Greater Than 1518 Bytes

You can use the SYSMAN utility or the LANCP utility to enable the use of jumbo frames. Directions for using both utilities follow.

The jumbo frame setting takes effect immediately, regardless of which utility you use. However, an application may check the maximum packet size only during initialization. For this reason, it may be necessary to stop and restart the application to effect the change.

Each time you change the frame size, a console message is displayed. If you change the frame size with SYSMAN, the following message is displayed:

```
%EWA0, Jumbo frames enabled per SYSGEN parameter LAN_FLAGS
```

3.19.3.2 SYSMAN Utility Directions

To enable jumbo frames by means of the SYSMAN utility:

1. Set bit 6 of the LAN_FLAGS system parameter, as shown in the following example:

   ```
   $ run sys$system:sysman
   SYSMAN> parameters use current
   SYSMAN> parameters set lan_flags %x40
   SYSMAN> parameters write active
   SYSMAN> parameters write current
   SYSMAN> exit
   ```

   The notation “%x” instructs the operating system that the following number is hexadecimal.

   Note

   The setting for bit 6 should be OR’d with any existing bits set in LAN_ FLAGS.

2. Insert or edit the following line in SYS$SYSTEM:MODPARAMS.DAT:

   ```
   lan_flags = %x40 ! enable jumbo frames on DEGPA
   ```

You can turn off the jumbo frame setting by clearing bit 6 and then editing the same line in SYS$SYSTEM:MODPARAMS.DAT.

3.19.3.3 LANCP Utility Directions

To enable jumbo frames by means of the LANCP utility:

1. Issue the following command:

   ```
   $ mc lancp set dev devname/jumbo_frames
   ```

2. Insert the previous command into SYS$SYSTEM:SYSTARTUP_VMS.COM to enable jumbo frames before applications are started.

To turn off jumbo frame support, replace `jumbo_frames` with `nojumbo_frames` and make the appropriate edit to the SYS$SYSTEM:SYSTARTUP_VMS.COM file.

3.20 PCSI Qualifiers

Beginning with OpenVMS Version 7.2, the POLYCENTER Software Installation (PCSI) utility has two new qualifiers that you can use with a number of PRODUCT commands:

Qualifier	Description
/KIT_ATTRIBUTES	Allows you to select kits by kit type or kit format, or both.
/SPAN_VERSIONS	Allows you to specify a range of versions using the keywords ABOVE, BELOW, MAXIMUM, and MINIMUM.

3.21 RMS CONVERT Utility Enhancements

For OpenVMS Version 7.2, the CONVERT utility has had several enhancements added to it. These improved features are described in the following paragraphs. For more detailed information, see the *OpenVMS Record Management Utilities Reference Manual* and the *OpenVMS Utility Routines Manual*.

The CONVERT utility has been modified to eliminate a previous design constraint in which the output file would temporarily become vulnerable to user access during the exchange of the file between CONVERT and the SORT32 utility. This would occur during the FAST load processing of the secondary keys of the file.

Previous versions of CONVERT required unlocking of the output file whenever it passed control to the SORT32 utility for performing sort operations. This provided an opportunity in which the output file could be locked by a user application, causing the file to become inaccessible to CONVERT. This could result in a premature exit of the convert process, leaving some of the secondary keys in an uninitialized state. This failure was most prevalent when the output file name was the same as the production file name.

The CONVERT modifications were made with the goal of not impacting performance or increasing disk space usage by default. In all test cases, the CONVERT performance has been either equal to or better than previous versions of CONVERT while maintaining constraints on the disk space required for the work files. A new qualifier, /SECONDARY, has been added that allows a user to override the latter default constraint to improve performance at the expense of disk space.

3.22 SYSMAN RESERVED_MEMORY: New and Modified Subcommands and Qualifiers

If you want to reserve a block of contiguous page frame numbers (PFNs), you can use new and modified SYSMAN RESERVED_MEMORY subcommands to do so. The following sections discuss:

- Changes to existing subcommands and qualifiers
- New subcommands and qualifiers that have been added

3.22.1 New and Changed Subcommands

The following sections lists new and modified qualifiers of existing RESERVED_ MEMORY subcommands.

New Qualifiers /SYSGBL and /[NO]GLOBAL

The subcommands ADD, FREE, REMOVE, and SHOW accept these new qualifiers:

/SYSGBL	Specify this qualifier to indicate that the reservation is for a system global memory resident section.
/GLOBAL_SECTION /NOGLOBAL_SECTION	Specify /NOGLOBAL_SECTION to indicate that the memory qualifier is for a privileged application rather than for a group or system global section.

Changes Related to the /GROUP Qualifier

You cannot use the /GROUP qualifier with either /SYSGBL or /NOGLOBAL_ SECTION.

Improved Ease of Use of /ZERO and /ALLOCATE Qualifiers

In OpenVMS Version 7.1, you could specify /ZERO only if you also specified /ALLOCATE. The new behavior is the following:

- /ZERO implies /ALLOCATE.
- /NOALLOCATE implies /NOZERO.
- /ZERO is incompatible with /NOALLOCATE.

This change has no effect on existing command procedures because all commands that were valid in Version 7.1 behave exactly as they did in Version 7.1 because /GLOBAL_SECTION is now the default.

3.22.2 New MODIFY Subcommand and Qualifiers

MODIFY	Allows you to modify an existing entry in the reserved memory registry file.
Format:	`RESERVED_MEMORY MODIFY gs_name`

3.23 System Parameters

OpenVMS Alpha Version 7.2 introduces several new system parameters. Table 3–12 lists the parameters and their functions. For additional information on OpenVMS system parameters, refer to online help and the *OpenVMS System Management Utilities Reference Manual*.

Table 3–12 New System Parameters

Parameter	Function
ARB_SUPPORT	A number of COMPAQ and third-party products require that OpenVMS maintain the pre-Version 7.2 Access Rights Block (ARB) and related kernel-mode data structures in parallel with the data structures that have replaced the ARB; this requirement continues because many products have not yet been modified to operate correctly with the new per-thread security Persona Security Block (PSB) data structures.
MAXBOBS0S1	On Alpha systems, MAXBOBS0S1 defines the maximum amount of 32-bit system space, measured in pagelets, that can be used for windows to buffer objects. MAXBOBS0S1 is a DYNAMIC parameter.
MAXBOBS2	On Alpha systems, MAXBOBS2 defines the maximum amount of 64-bit system space, measured in pagelets, that can be used for windows to buffer objects. MAXBOBS2 is a DYNAMIC parameter.
MPDEV_D1	MPDEV_D1 is reserved for Compaq use only.
MPDEV_ENABLE	MPDEV_ENABLE controls the formation of multipath sets. When MPDEV_ENABLE is ON (1), the formation of multipath sets is enabled. Setting this parameter to OFF (0) disables the formation of additional multipath sets; path failover, however, continues on existing multipath sets. The default value is ON (1). MPDEV_ENABLE is a DYNAMIC parameter.
MPDEV_LCRETRIES	MPDEV_LCRETRIES specifies the number of times the system retries locally connected paths to a SCSI device before moving on to local unconnected paths or to the MSCP-served path to the device. The valid range is 1 through 256. MPDEV_LCRETRIES is a DYNAMIC parameter.
MPDEV_POLLER	MPDEV_POLLER controls the polling of paths to multipath set members. Setting this parameter to ON (1) allows early detection of errors on otherwise inactive paths. If a path becomes unavailable or returns to service, an OPCOM message notifies the system manager. If MPDEV_POLLER is set to OFF (0), multipath polling is disabled. The default value is ON (1). MPDEV_POLLER is a DYNAMIC parameter.
MPDEV_REMOTE	MPDEV_REMOTE controls whether the MSCP served path to a SCSI device can become a member of a multipath set. Setting the parameter to ON (1) allows this membership. If MPDEV_REMOTE is set to OFF (0), only local paths to a SCSI device can be used to form additional multipath sets; path failover, however, continues to include remote members of existing multipath sets. The default value is ON (1). MPDEV_REMOTE is a DYNAMIC parameter.

(continued on next page)

Table 3–12 (Cont.) New System Parameters

Parameter	Function
POWEROFF	POWEROFF enables or disables software requests to the console firmware to remove power from the system. This parameter should normally be turned ON (1) to allow software to make power-off requests. However, POWEROFF can be set to OFF (0) to disable software power-off requests. If firmware or hardware support for the power-off request is not implemented, the shut-down procedure leaves the system halted, but fully powered. POWEROFF is a DYNAMIC parameter.
NPAG_BAP_MIN_PA	On Alpha systems, NPAG_BAP_MIN_PA specifies the lowest physical address allowed within bus addressable pool.
VCC_MAX_CACHE	On Alpha systems, VCC_MAX_CACHE is reserved for Compaq use only.
VCC_MAX_LOCKS	On Alpha systems, VCC_MAX_LOCKS is reserved for Compaq use only.
VCC_MINSIZE	On VAX systems, VCC_MINSIZE sets the lower limit in pages of memory used by virtual I/O cache.
VCC_WRITEBEHIND	On Alpha systems, VCC_WRITEBEHIND is reserved for Compaq use only.
VCC_WRITE_DELAY	On Alpha systems, VCC_WRITE_DELAY is reserved for Compaq use only.
WLKSYSDSK	On Alpha systems, WLKSYSDSK is used by various bootstrap components to determine if the system disk should be treated as though it is write-locked.

3.24 Tape Density Support Enhanced (Alpha Only)

In versions of OpenVMS prior to Version 7.2, the range of densities that users were able to set for magnetic tape devices was limited. With OpenVMS Version 7.2, that range has been extended to include any density that a specific tape drive supports. Because of this enhancement, exchanging tapes among tape drives with different default settings for density is much easier.

Using multiple tape densities is discussed in the *OpenVMS System Manager's Manual*.

You can set densities using the following DCL commands:

- $ INITIALIZE
- $ SET MAGTAPE (for tapes only)

Refer to the *OpenVMS DCL Dictionary* for details about using the /DENSITY qualifier with these DCL commands.

You can also set densities using the following system management utilities:

- $ MOUNT (with /FOREIGN qualifier for mounted tapes)
- $ BACKUP

Refer to the *OpenVMS System Management Utilities Reference Manual* for details about using the /DENSITY qualifier with these utilities.

Multiple tape density support is provided by changes in the QIO interface. These changes are guided by device/density tables in system libraries and the corresponding class drivers. This enhancement functions with tape drives that support multiple tape density switching via the standard MODE_SENSE and MODE_SELECT mechanisms. The QIO interface uses the information in the libraries and drivers to identify and match valid densities and compressions with tape drives.

For more information about enhanced tape density support, refer to the *OpenVMS I/O User's Reference Manual*.

3.25 TCP/IP Services for OpenVMS

With OpenVMS Version 7.2, the DIGITAL TCP/IP Services for OpenVMS Version 5.0 product replaces Version 4.2 (also known as UCX). Version 4.2 is no longer supported, but will remain on your OpenVMS system after the upgrade to Version 5.0. Version 5.0 completes the change initiated several releases ago when the product name changed from ULTRIX Connection (UCX) to DIGITAL TCP/IP Services for OpenVMS.

The identifier UCX is replaced with TCP/IP in the following items:

- Registered product facility code
- Management command prompt
- All messages, examples, and banners
- All product file names and databases
- All logical names
- All associated product documentation

OpenVMS Version 7.2 provides backward compatibility for all UCX logical names and support for UCX> commands.

New TCP/IP features include:

- A new, reliable, commercial-ready kernel, ported from the DIGITAL UNIX Version 4.0D IPv4 protocol stack.

  This kernel incorporates several performance enhancements either developed or extended by Compaq.

- A new option for dynamic routing—the gateway routing (GATED) daemon.

  GATED supports the following protocols:

  - Routing Information Protocol (RIP) Version 1 (RFC 1058) and RIP Version 2 (RFC 1388)

    RIP is a commonly used interior protocol that selects the route with the lowest metric (hop count) as the best route.

  - Open Shortest Path First (OSPF) Version 2 (RFC 1583)

    Another interior routing protocol, OSPF is a link-state protocol (shortest path first) and better suited than RIP for use in complex networks with many routers.

  - Exterior Gateway Protocol (EGP) (RFC 904)

EGP exchanges reachability information between autonomous systems. An autonomous system is usually defined as a set of routers under a single administration, using an interior gateway protocol and common metric to route packets. Autonomous systems use exterior routing protocols to route packets to other autonomous systems.

- Border Gateway Protocol (BGP) (RFC 1163 RFC 1267 RFC 1654)

  Like EGP, BGP exchanges reachability information between autonomous systems, but supports nonhierarchical topologies. BGP uses path attributes to provide more information about each route. Path attributes can include, for example, administrative preferences based on political, organizational, or security considerations.

- Router Discovery (RFC 1256)

  This protocol is used to inform hosts of the availability of hosts it can send packets to and is used to supplement a statically configured default router.

- A new management interface based on the following UNIX management utilities:

UNIX Utility	Function
`ifconfig`	Configures or displays network interface parameters, redefines an address for a particular interface, or sets options such as an alias list, broadcast address, or access filter.
`netstat`	Displays network statistics of sockets, data link counters, specified protocols or aliases, network interfaces, and a host's routing table.
`sysconfig`	Displays and maintains the network subsystem attributes.
`route`	Manually manipulates the routing table.
`arp`	Controls and displays ARP tables.

Important

If you are not familiar with these UNIX utilities, you should continue to use the OpenVMS-style management commands provided with the product and available at the TCPIP> prompt.

- New support for Dynamic Host Configuration Protocol (DHCP). DHCP provides a centralized approach to the configuration and maintenance of IP address space. It lets the system manager provide dynamic allocation of IP addresses from a single OpenVMS host.
- A new implementation of DNS/BIND based on the Internet Software Consortium's (ISC) BIND 8.1.
- A new method for cluster load balancing.
- New implementations of NTP and SNMP.
- A new TELNET port driver QIO programming interface.

Other changes for OpenVMS Version 7.2 include the following:

- Enhanced messages

- Obsolescense of some management commands
- Timezone configuration changes
- Enhanced socket-based application programming interface
- FTP enhancements, including support for Extended File Specifications
- NFS and SMTP changes

For more information about these features and changes, refer to the release notes available in the DIGITAL TCP/IP Services for OpenVMS Version 5.0 kit.

For information about installing DIGITAL TCP/IP Services for OpenVMS, refer to the manual *DIGITAL TCP/IP Services for OpenVMS Installation and Configuration*.

3.26 Ultra SCSI Support

Ultra SCSI was invented by DIGITAL and subsequently standardized by the ANSI SCSI committee. Ultra SCSI incorporates several improvements over its predecessor, Fast SCSI, including an increase in the maximum transfer rate on the SCSI bus from 10 MHz to 20 MHz. For a wide Ultra SCSI bus, this means an increase in maximum bus bandwidth from 20 MB/s to 40 MB/s.

OpenVMS Ultra SCSI support was first introduced in the OpenVMS Alpha Version 7.1-1H1 hardware release. OpenVMS Version 7.2 extends Ultra SCSI support to multihost configurations, as described in Section 3.16.9.

3.26.1 Coexistence of Ultra SCSI Devices with Other SCSI Devices

Ultra SCSI devices can be used on the same SCSI bus as non-Ultra SCSI devices. For example, Ultra SCSI peripherals such as the RZ1DB-VW can be used at non-Ultra speeds with the KZPSA adapter. Furthermore, non-Ultra SCSI peripherals such as the RZ29B can be used with the KZPBA Ultra SCSI host adapter.

3.26.2 Ultra SCSI Devices Supported by OpenVMS

For information about all Ultra SCSI devices supported by OpenVMS and how to configure them, refer to *StorageWorks UltraSCSI Configuration Guidelines*, order number EK–ULTRA–CG. You can also obtain information about StorageWorks Ultra SCSI products from their Web site, which is periodically updated:

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

3.27 Year 2000 Readiness

OpenVMS Version 7.2 is fully Year 2000 ready. This version of OpenVMS includes all the Year 2000 enhancements that shipped in the Year 2000 kits for OpenVMS Version 7.1 and 7.1–1H1.

The OpenVMS Year 2000 enhancements are the result of a rigorous and comprehensive analysis of the entire OpenVMS operating system, including simulations of the transition to the Year 2000 and beyond, and extensive OpenVMS testing. These Year 2000 modifications affect only a few older and rarely used components.

You can find links to OpenVMS Year 2000 release notes and other information about testing your own environment for the Year 2000 on the OpenVMS Year 2000 web site:

http://www.openvms.digital.com/openvms/products/year-2000/

OpenVMS Version 7.2 conforms to the Year 2000 DIGITAL Product Warranty. For information on this special warranty, see the following World Wide Web page:

http://www.digital.com/year2000/warranty.asp

Although OpenVMS is ready for the Year 2000, you must ensure that the environment in which OpenVMS operates is also ready. Based on our investigations and testing, Compaq expects that most Year 2000-related problems will occur primarily in locally developed layered applications. Therefore, it is important to start evaluating your applications and environments as soon as possible.

4
Programming Features

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

4.1 Adapter Support (Alpha Only)

The following sections describe the new adapters supported for OpenVMS Alpha Version 7.2.

4.1.1 ATMWORKS 351 Adapter

The ATMWORKS 351 adapter is a high-performance, 155 Mb/s, full-duplex ATM adapter that enables systems with peripheral component interconnect (PCI) slots to communicate over an ATM network. The ATMWORKS 351 driver is supported by the SYS$ATMWORKS351 port driver which has a device name HW*cu*, where *c* is the controller and *u* is the unit number, as for example, HWA0.

For more detailed information, see the *OpenVMS I/O User's Reference Manual*.

4.1.2 DE500-BA/DE504-BA Adapters

OpenVMS Alpha Version 7.2 provides run-time and boot support for the DE500-BA and DE504-BA adapters. The DE500-BA and DE504-BA are single-slot adapters with direct interfaces to the 32-bit PCI Ethernet Network Interface Card (NIC). The DE500-BA contains a single port, and the DE504-BA contains four ports. Each port is capable of running Fast Ethernet (100 Mb/s) and 10 Mb/s standard Ethernet protocols. Both adapters support full- and half-duplex modes at either speed for configuration flexibility that allows a single adapter to communicate with Fast or standard Ethernet systems. Also, both adapter's NIC also includes support for IEEE 802.3u autonegotiation.

The DE500-BA uses an 8-position modular connector to connect to hubs, switches, or to other NICs. This connector, commonly referred to as an RJ-45, uses two pairs of data grade 100-ohm category 5 UTP or 150-ohm STP-A cable supporting a maximum distance of 100 meters.

4.1.3 DE500-FA Adapter

OpenVMS Alpha Version 7.2 provides run-time and boot support for the DE500-FA PCI FastEthernet Network Interface Card (NIC). The DE500-FA is capable of 100 Mb operation in half or full-duplex mode. This NIC supports the 100Base-FX physical layer described in clause 26 of the IEEE 802.3u specification. The 100Base-FX is designed to operate over two strands of 62.5/125 graded index multimode optical fiber cable. A maximum distance of 412 meters is supported in half-duplex mode and 2000 meters in full-duplex mode. The DE500-FA uses an SC fiber optic connector. The 100Base-FX does not support autonegotiation or 10 Mb operation.

4.1.4 DEGPA-SA Adapter

OpenVMS Version 7.2 provides run-time support for the DEGPA-SA PCI Gigabit Ethernet Network Interface Card (NIC). The DEGPA-SA conforms to the IEEE 802.3z Gigabit Ethernet standard running over 1000BASE-SX fiber optic cabling. It can run point-to-point to another Gigabit Ethernet adapter or to a Gigabit Ethernet hub/switch. For more information about Gigabit Ethernet support, see Section 3.19.

4.2 Clusterwide Logical Names

Clusterwide logical names are an extension to the existing logical name support in OpenVMS. They are available on OpenVMS Alpha and on OpenVMS VAX.

This section provides information regarding the use of clusterwide logical names in applications. For general information about clusterwide logical names, including how to create them at the DCL level, see Section 3.16.2. For more information about using logical names in applications, refer to the *OpenVMS Programming Concepts Manual*.

4.2.1 New Clusterwide Attributes for $TRNLNM System Service

Two new attributes have been added to the $TRNLNM system service:

- LNM$V_CLUSTERWIDE
- LNM$M_INTERLOCKED

LNM$V_CLUSTERWIDE is an output attribute to be returned in the itemlist if you asked for the LNM$_ATTRIBUTES item for a logical name that is clusterwide.

LNM$M_INTERLOCKED is an **attr** argument bit that can be set to ensure that any clusterwide logical name modifications in progress are completed before the name is translated. LNM$M_INTERLOCKED is not set by default. If your application requires translation using the most recent definition of a clusterwide logical name, use this attribute to ensure that the translation is stalled until all pending modifications have been made.

On a single system, when one process modifies the shareable part of the logical name database, the change is visible immediately to other processes on that node. Moreover, while the modification is in progress, no other process can translate or modify shareable logical names.

In contrast, when one process modifies the clusterwide logical name database, the change is visible immediately on that node, but it takes a short time for the change to be propagated to and made on other nodes. By default, translations of clusterwide logical names are not stalled. Therefore, it is possible for processes on different nodes to translate a logical name and get different equivalence names when modifications are in progress.

The use of LNM$M_INTERLOCKED guarantees that your application will receive the most recent definition of a clusterwide logical name.

4.2.2 New Clusterwide Attribute for $GETSYI System Service

One new clusterwide attribute, SYI$_CWLOGICALS, has been added to the $GETSYI system service. When you specify SYI$_CWLOGICALS, $GETSYI returns the number 1 if the clusterwide logical name database has been initialized on the CPU, or the value 0 if it has not been initialized. Because this number is a Boolean value (1 or 0), the buffer length field in the item descriptor should specify 1 (byte). On a nonclustered system, the value of SYI$_CWLOGICALS is always 0.

4.2.3 Creating Clusterwide Tables with the $CRELNT System Service

When creating a clusterwide table, the $CRELNT requester must supply a table name. OpenVMS does not supply a default name for clusterwide tables because the use of default names enables a process without the SYSPRV privilege to create a shareable table.

4.3 COM for OpenVMS

COM (Component Object Model) is a technology from Microsoft® that allows developers to create distributed network objects. Digital Equipment Corporation and Microsoft jointly developed the COM specification. First released as NetOLE (Network Object Linking and Embedding) and then renamed DCOM (Distributed COM), the COM specification now includes network objects. COM for OpenVMS is an implementation of the Microsoft code that supports the COM draft standards.

A developer might implement COM applications on OpenVMS in the following ways:

- For existing OpenVMS applications and data, a developer can encapsulate an OpenVMS application as a COM object. A Windows COM client can then interact with the existing OpenVMS application and data. This allows business-critical applications and data to stay on the OpenVMS system while making the OpenVMS data and code accessible through a Windows application.
- For existing COM applications on other operating systems, a developer can port COM objects to OpenVMS to take advantage of specific OpenVMS features. This allows you to leave other COM objects associated with the application unchanged on their current platforms, and lets you port only the objects you need to your OpenVMS system. You can then revise the ported code to maximize OpenVMS benefits and at the same time, minimize the change to the overall application.
- For new OpenVMS applications, a developer can create a COM server on OpenVMS. COM client applications on Windows NT systems, on other OpenVMS systems, or on other COM-enabled operating systems can access this OpenVMS COM server.

To implement COM on OpenVMS, Compaq has made the following changes to the OpenVMS operating system:

- Added COM APIs to OpenVMS

  Compaq has added a set of Microsoft COM APIs to OpenVMS. These APIs allow developers to write and debug COM objects on OpenVMS systems. (For complete details about COM for OpenVMS development, see the *OpenVMS Connectivity Developer Guide* included with the COM for OpenVMS developer kit.)

- Added an OpenVMS Registry

  COM applications can add, read, change, and delete the OpenVMS Registry contents. You can view and edit the contents of the OpenVMS Registry either from OpenVMS (using the `REG$CP` utility or the `$REGISTRY` system service) or from Windows® NT™ (using `RegEdit` or `RegEdt32`).

- Extended OpenVMS security to enable secure connections between Windows NT and OpenVMS

  These enhancements include support for common user authentication and authenticated RPC with NTLM security. In addition, Compaq has moved some DCE RPC functions into the base operating system to provide limited support for Microsoft RPC calls.

- Added application event logging on OpenVMS

  Applications write system, security, and application activity information to the OpenVMS event logs. You can view the contents of the event logs on OpenVMS either from OpenVMS (using the PATHWORKS Admin utility) or from Windows NT (using the Windows NT Event Viewer).

4.3.1 COM for OpenVMS Delivery

COM for OpenVMS will ship with the OpenVMS operating system. It is licensed as follows:

- The run-time version of COM for OpenVMS is licensed as part of OpenVMS Version 7.2.
- The COM for OpenVMS developer kit has a separately orderable license.

4.3.2 COM for OpenVMS Security

COM for OpenVMS security will be implemented in two phases. The following sections describe the phases.

Phase 1: COM Version 1.0 for OpenVMS (without authentication)

In this phase, a COM for OpenVMS process executes with an OpenVMS security identity only; OpenVMS does not authenticate COM requests from Windows NT clients or process any Windows NT credentials.

An OpenVMS system manager can set the COM for OpenVMS security identities of a COM server process in the following ways:

- Execute the process with a `DCOM$GUEST` identity.

  `DCOM$GUEST` is a nonprivileged account created by COM for OpenVMS during installation. (This is the default action.)

- Assign an OpenVMS account to a specific COM for OpenVMS application.

  In this case, the system manager creates a OpenVMS account that has privileges, rights, and restrictions as defined by the person creating the account. COM servers that execute from this account are restricted to the OpenVMS security context of the account.

Because COM Version 1.0 for OpenVMS does not authenticate remote users, COM for OpenVMS accepts and processes client requests as if authentication had taken place. Although less secure than a full NTLM implementation, COM Version 1.0 for OpenVMS minimizes the security risk by using the OpenVMS accounts to execute servers. COM Version 1.0 for OpenVMS enforces security on a processwide basis; as a result, per-method security is not available.

Because the COM process has no associated NT credentials and no authentication mechanism exists in COM Version 1.0 for OpenVMS, Windows NT systems treat the outbound requests to Windows NT systems as unauthenticated. Windows NT systems that run COM server processes for COM for OpenVMS client applications must allow access to everyone for the specific server applications.

When full NTLM authentication (COM Version 1.1 for OpenVMS) is available, Compaq will add another option: *client access*. This option allows the COM for OpenVMS server process to execute in the security context of the requesting Windows NT client. The COM for OpenVMS server process includes Windows NT credentials that OpenVMS can use for Registry access and outbound COM requests.

COM Version 1.0 for OpenVMS software requirements

COM Version 1.0 for OpenVMS does not use the NTLM security features (tactical security, SSPI, and authenticated RPC). COM Version 1.0 for OpenVMS does not require or use Advanced Server for OpenVMS (formerly the PATHWORKS server). Advanced Server for OpenVMS is required if you want to connect from a Windows NT system and access the OpenVMS Registry. Event logging is part of PATHWORKS and is not available in COM Version 1.0 for OpenVMS.

Phase 2: COM Version 1.1 for OpenVMS (with NTLM authentication)

In this phase, COM for OpenVMS processes OpenVMS security identities, authenticates COM requests from Windows NT clients, and processes Windows NT credentials. This is the full implementation of NTLM (NT LAN Manager) security for COM for OpenVMS.

COM Version 1.1 for OpenVMS software requirements

COM Version 1.1 for OpenVMS uses the NTLM security features (tactical security, SSPI, and authenticated RPC). COM Version 1.1 for OpenVMS requires Advanced Server for OpenVMS (formerly the PATHWORKS server). COM Version 1.1 for OpenVMS enables event logging.

4.3.2.1 Summary of Security Implementation Differences

The OpenVMS Registry can control access to OpenVMS Registry keys in the following two ways:

- Through NTLM security

  NTLM security uses NT credentials to control access to registry keys and can be used to control access to specific keys.

- Through OpenVMS security

  OpenVMS security uses OpenVMS privileges and rights identifiers to control access to the OpenVMS Registry database and *cannot* be used to control access to specific keys.

COM applications require read access to the COM registry keys and COM developers require read and write access to the COM registry keys. In COM Version 1.0 for OpenVMS (without authentication), COM for OpenVMS does not have NT credentials; as a result, COM Version 1.0 for OpenVMS uses OpenVMS security to control access to the OpenVMS Registry. Because you cannot control this OpenVMS Registry access on a per-key basis, you must grant all COM application read access to the entire OpenVMS Registry and you must grant all COM developers write access to the entire OpenVMS Registry. This means that the entire OpenVMS Registry, including the PATHWORKS portion not used by COM for OpenVMS, is accessible to COM applications and developers.

In COM Version 1.1 for OpenVMS (with NTLM authentication), COM for OpenVMS will use NT credentials to control access to specific OpenVMS Registry keys and remove the OpenVMS privileges and rights identifiers. This will protect those parts of the OpenVMS Registry that are not used by COM for OpenVMS.

4.4 Common File Qualifier Routines

OpenVMS Version 7.2 contains the UTIL$CQUAL routines, which allow you to parse the command line for qualifiers related to certain file attributes, and to match files you are processing against the selected criteria retrieved from the command line. The utility routines allow a user to select files and perform terminal I/O without the application writer performing these explicitly.

The common file qualifier routines begin with the characters UTIL$CQUAL. Your program calls these routines using the OpenVMS Calling Standard. When you call a UTIL$CQUAL routine, you must provide all the required arguments. Upon completion, the routine returns its completion status as a condition value.

The following table lists the common file qualifier routines.

Table 4–1 UTIL$CQUAL Routines

Routine Name	Description
UTIL$CQUAL_FILE_PARSE	Parses the command line for file qualifiers, and obtains associated values. Returns a context value that is used when calling the matching and ending routines.
UTIL$CQUAL_FILE_MATCH	Compares the routine file input to the command line data obtained from the parse routine call.
UTIL$CQUAL_FILE_END	Deletes all virtual memory allocated during the command line parse routine call.
UTIL$CQUAL_CONFIRM_ACT	Prompts a user for a response from SYS$COMMAND.

4.4.1 Using the Common File Qualifier Routines

Follow these steps to use the common file qualifier routines:

1. Call UTIL$CQUAL_FILE_PARSE to parse the command line for the common file qualifiers.
2. Call UTIL$CQUAL_FILE_MATCH for each checked file. UTIL$CQUAL_ FILE_MATCH returns an indication that the file is, or is not, to be processed.
3. Call UTIL$CQUAL_FILE_END to release the virtual memory held by the common file qualifier package.

You may optionally call UTIL$CQUAL_CONFIRM_ACT to ask for user confirmation without calling the other common qualifier routines.

For more information about the common file qualifier routines, refer to the *OpenVMS Utility Routines Manual*.

4.5 DECthreads

This section contains information about new DECthreads features for OpenVMS Version 7.2.

4.5.1 Yellow Zone Stack Overflow (Alpha only)

DECthreads now provides a special protected area of memory between the usable stack region and the guard page. When accessed, the VMS exec un-protects this memory and a stack overflow exception is delivered to the user thread. This feature enables applications to catch stack overflow conditions and attempt recovery or graceful termination.

4.5.2 Read-Write Locks (Alpha and VAX)

DECthreads now supports use of multiple readers, single writer locks, also known as read-write locks, allowing many threads to have simultaneous read-only access to data while allowing only one thread to have write access at any given time. Such locks are typically used to protect data that is read more frequently than written.

As part of this support, the following new routines have been added to the DECthreads POSIX 1003.1c library:

```
pthread_rwlock_destroy()
pthread_rwlock_init()
pthread_rwlock_rdlock()
pthread_rwlock_tryrdlock()
pthread_rwlock_trywrlock()
pthread_rwlock_unlock()
pthread_rwlock_wrlock()
pthread_rwlockattr_destroy()
pthread_rwlockattr_init()
```

4.5.3 Threads Debugging Enhancements (Alpha and VAX)

DECthreads has improved its debugging support, allowing Compaq and third-party debuggers to better integrate with DECthreads. The 'pthread' debugging interface, previously only available via SDA is now available from the OpenVMS Debugger as well.

4.6 DIGITAL DCE Remote Procedure Call (RPC) Functionality

Note

Information on Microsoft's NT Lan Manager (NTLM) is provided as a preview of functionality that will be available in a future version of Digital DCE for OpenVMS (Alpha only). This advanced documentation will help you in future planning.

Beginning with OpenVMS Version 7.2, Remote Procedure Call (RPC) functionality is integrated into the operating system. RPC provides connectivity between individual procedures in an application across heterogeneous systems in a transparent way. This functionality uses the DIGITAL Distributed Computing Environment (DCE) RPC, but also provides the ability to use the Microsoft RPC Application Programming Interface (API) as well as the DIGITAL DCE RPC API. Using RPC, an application can interoperate with either DIGITAL DCE or Microsoft RPC applications. If security is required, you can use either Microsoft's

NTLM security or (by installing the full DIGITAL DCE Run-Time Kit) DCE Security.

The RPC daemon allows client/server RPC applications to register their specific endpoints. Registering endpoints allows remote clients of the application to find the server application's entry point on the system.

The DIGITAL DCE Application Developer's Kit (separately licensed) is required to develop RPC applications, but the resulting applications may be installed on any OpenVMS Version 7.2 system, or any supported OpenVMS system where the full DIGITAL DCE Run-Time Kit is installed. The DIGITAL DCE Run-Time Kit, although shipped on the OpenVMS CD–ROM, is licensed with the OpenVMS operating system and does not require the purchase of additional software licenses.

4.6.1 Starting and Stopping RPC

The RPC daemon can be started or stopped with the following new command files:

DCE$RPC_STARTUP.COM
DCE$RPC_SHUTDOWN.COM

These command files are located in the directory SYS$COMMON:[SYSMGR].

To start the RPC daemon, execute the DCE$RPC_STARTUP.COM procedure. The following option may be specified:

[NO]CONFIRM

Turns user prompting on or off. CONFIRM is the default.

To stop the RPC daemon, execute the DCE$RPC_SHUTDOWN.COM procedure. The following options may be specified in any order:

[NO]CONFIRM

Turns user prompting on or off. CONFIRM is the default.
CLEAN

Deletes all entries from the RPC endpoint database.

Note

Do *not* stop the RPC daemon if any DCE components or RPC applications are running on the system.

4.6.2 Managing RPC Endpoints

RPC endpoints can be managed using the RPC Control Program (RPCCP). To invoke this utility, enter the following command:

```
$ RUN SYS$SYSTEM:DCE$RPCCP.EXE
```

You can type `help` at the system prompt for information about parameter usage.

4.6.3 Limiting RPC Transports

The RPC daemon can limit what protocols will be used by RPC applications. To restrict the protocols that can be used, set the logical name RPC_SUPPORTED_PROTSEQS to contain the valid protocols. Valid protocols are ncadg_ip_udp, ncacn_ip_tcp, and ncacn_dnet_nsp. Separate each protocol with a colon. For example:

```
$ define RPC_SUPPORTED_PROTSEQS "ncacn_ip_tcp:ncacn_dnet_nsp"
```

This definition prevents RPC applications from registering endpoints that use TCP/UDP.

4.6.4 For More Information

Refer to the *OpenVMS Version 7.2 Release Notes* for important information for existing users of DIGITAL DCE for OpenVMS.

For additional information about DIGITAL DCE for OpenVMS, use online help or refer to the following documentation:

- *Digital DCE for OpenVMS VAX and OpenVMS Alpha Installation and Configuration Guide*
- *Digital DCE for OpenVMS VAX and OpenVMS Alpha Product Guide*
- *Digital DCE for OpenVMS VAX and OpenVMS Alpha Reference Guide*

4.7 Fast I/O and Buffer Objects for Global Sections (Alpha Only)

As of OpenVMS Alpha Version 7.2, VLM applications can use Fast I/O for memory shared by processes through global sections. In prior versions of OpenVMS Alpha, buffer objects could be created only for process private virtual address space. Database applications where multiple processes share a large cache can now create buffer objects for the following types of global sections:

- Page file-backed global sections
- Disk file-backed global sections
- Memory-resident global sections

Buffer objects enable Fast I/O system services, which can be used to read and write very large amounts of shared data to and from I/O devices at an increased rate. By reducing the CPU cost per I/O request, Fast I/O increases performance for I/O operations. Fast I/O improves the ability of VLM applications, such as database servers, to handle larger capacities and higher data throughput rates.

For more information about how to use Fast I/O and buffer objects for global sections, see the *OpenVMS Alpha Guide to 64-Bit Addressing and VLM Features*.

4.8 Fast Path Support (Alpha Only)

Fast Path is an optional, high-performance feature designed to improve I/O performance. Fast Path creates a streamlined path to the device. Fast Path is of interest to any application where enhanced I/O performance is desirable. Two examples are database systems and real-time applications, where the speed of transferring data to disk is often a vital concern.

Using Fast Path features does not require source-code changes. Minor interface changes are available for expert programmers who want to maximize Fast Path benefits.

Beginning with OpenVMS Alpha Version 7.1, Fast Path supports disk I/O for the CIXCD and the CIPCA ports. These ports provide access to CI storage for XMI- and PCI-based systems. In Version 7.0, Fast Path supported disk I/O for the CIXCD port only.

Fast Path is not available on the OpenVMS VAX operating system.

For more information, see Table 1–1 and Section 4.21.5 in this manual, and refer to the *OpenVMS I/O User's Reference Manual*.

4.9 Fast Skip for SCSI Tape Drives

If you access your tape drive via your own user-written program, you can use a new modifier, IO$M_ALLOWFAST, to control the behavior of the IO$_SKIPFILE function.

In the past, OpenVMS has always skipped files on tape by using skip-record commands to perform the motion. This has essentially simulated skip-by-filemarks, and while this functions correctly, this type of skipping can be quite slow at times as compared to what a tape can achieve by using skip-by-filemarks commands.

In V7.2, OpenVMS allows the tape driver to be set up to use skip-by-filemarks where the user permits. Modern tape drives keep track of the end of logical data, so MKdriver can use these facilities to position correctly if these features exist on a tape drive. (If they do not, the old skip-by-records skipping is used.)

The skipping-by-filemarks, when used, will correctly sense the end-of-tape provided the tape ends in double EOF. Utilities such as Backup and file structured copy, which use ANSI formatted tape, need the end positioning to be set correctly, and they work fine with the new skip method. They already continue skip-file operations across null files if they exist.

Some third-party utilities may, however, depend on the documented behavior of stopping a skip-by-files on double EOF marks on tape. To accommodate this, OpenVMS assumes by default the use of the old skip-by-records form of tape motion.

Because this form of positioning may be as much as 100 times slower than skip-by-files, OpenVMS provides two additional features:

1. The magtape ACP and Backup uses an added function modifier in their IO$_SKIPFILE commands to tape (IO$M_ALLOWFAST) that permits them to use the skip-by-filemarks command. This modifier is also transmitted across TMSCP, if all systems are at the 7.2 level.
2. The SET MAGTAPE command has a new qualifier, /FAST_SKIP. For more information on this qualifier, see Section 3.8 in this manual, and refer to the *OpenVMS DCL Dictionary: N–Z*, and online help.

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

This section briefly describes new features and capablilities pertaining to the command line interface and the callable interface (SOR routines) of the OpenVMS Alpha high-performance Sort/Merge utility. This information is of interest to both general users and programmers.

For more information about these new features, and about using the OpenVMS Alpha high-performance Sort/Merge utility, refer to the *OpenVMS User's Manual* and the *OpenVMS Utility Routines Manual*.

4.10.1 High-Performance Sorting with Threads

Support for threads has been added to the high-performance Sort/Merge utility. This enables enhanced performance by taking advantage of multiple processors on an SMP configured system.

To obtain best performance when using the high-performance Sort/Merge utility on a multi-processor machine, your program should be linked with the /THREADS_ENABLE qualifier.

4.10.2 Indexed Sequential Output File Organization

Support for indexed sequential output file organization has been added to the high-performance Sort/Merge utility.

You may now specify the /INDEXED_SEQUENTIAL file organization qualifier.

4.10.3 Output File Overlay

Support for output file overlay has been added to the high-performance Sort /Merge utility.

You may now specify the /OVERLAY qualifier to overlay or write an output file to an existing empty file.

4.10.4 Statistical Summary Information

Partial support for statistical summary information has been added to the high-performance Sort/Merge utility.

You may now specify the /STATISTICS qualifier with the following fields:

- Records read/input (SOR$K_REC_INP)
- Records sorted (SOR$K_REC_SOR)
- Records output (SOR$K_REC_OUT)
- Input record length (SOR$K_LRL_INP)

4.11 Intra-Cluster Communication System Services

The new intra-cluster communication (ICC) system services, available on Alpha and VAX, form an application programming interface (API) for process-to-process communications. For large data transfers, the ICC system services are the highest-performance OpenVMS application communication mechanism, superior to standard network transports and mailboxes.

The ICC system services enable application program developers to create distributed applications with connections between different processes on a single system, or between processes on different systems within a single OpenVMS Cluster system.

The ICC system services do not require a network product. The communication uses memory or System Communication Services (SCS).

The ICC system services:

- Allow the creation of both client and server processes
- Maintain a simple registry of servers and services
- Manage security of the server process namespace and access to server processes

- Establish connections between these processes and transmit data between these processes
- Provide 64-bit buffer and address support

4.11.1 ICC Benefits

The ICC system services provide the following benefits:

- An easy-to-use conventional system service interface for interprocess communications within a cluster
- An interface usable for communications between processes within a single, nonclustered node
- An interface callable from all modes and from execlets as well as from images
- An easy-to-use interface giving access to high-speed interconnects such as Memory Channel
- An interface independent of the installation of any networking product
- An interface usable for nonprivileged clients and authorized (but not necessarily privileged) servers

4.11.2 ICC System Services

The new system services enable application program developers to create connections between processes on the same or different systems within a single OpenVMS Cluster system.

These services are as follows:

- Open Association: $ICC_OPEN_ASSOC
- Close Association: $ICC_CLOSE_ASSOC
- Connect: $ICC_CONNECT and $ICC_CONNECTW
- Accept: $ICC_ACCEPT
- Reject: $ICC_REJECT
- Disconnect: $ICC_DISCONNECT and $ICC_DISCONNECTW
- Transmit Data: $ICC_TRANSMIT and $ICC_TRANSMITW
- Receive Data: $ICC_RECEIVE and $ICC_RECEIVEW
- Transceive Data: $ICC_TRANSCEIVE and $ICC_TRANSCEIVEW
- Reply: $ICC_REPLY and $ICC_REPLYW

Refer to the *OpenVMS System Services Reference Manual: GETQUI–Z* for additional information on the ICC system services.

4.11.3 Programming with ICC

Refer to the *OpenVMS Programming Concepts Manual* for information on programming with ICC.

4.11.4 ICC System Management and Security

Refer to the *OpenVMS System Manager's Manual* for information on ICC system management and security.

4.12 Java Development Kit (Alpha Only)

The Java Development Kit (JDK) is shipped with the OpenVMS operating system. This kit can be used to develop and run Java applets and programs on OpenVMS Alpha systems.

The JDK for OpenVMS systems contains a just-in-time (JIT) compiler. The JIT compiler provides on-the-fly compilation of your application's Java byte-codes and runtime calls into native Alpha machine code. This results in significantly faster execution of your Java application compared with running it using the Java interpreter. The JIT runs by default when you enter the JAVA command.

The JDK implements Java threads on top of native (POSIX) threads. This allows different Java threads in your application to run on different processors, provided that you have a multiprocessor machine. It also means that your Java application will run properly when linked with native methods or native APIs (such as DCE) that are also implemented using POSIX threads.

For more information, see the Java documentation in the following directory on your OpenVMS Alpha system where the JDK is installed:

```
SYS$COMMON:[SYSHLP.JAVA]INDEX.HTML
```

4.13 Kernel Threads Enhancements (Alpha Only)

OpenVMS Alpha Version 7.2 includes the following new kernel threads capabilities:

- Capacity for a larger number of kernel threads per process
- Ability to change the base priorities of all kernel threads with a single call
- Ability to detect a thread stack overflow

4.13.1 Increased Number of Kernel Threads Per Process

In the initial release of the kernel threads support, OpenVMS allowed a maximum of 16 kernel threads per process. This enabled an application to have threads executing on up to 16 CPUs at one time. With OpenVMS Alpha Version 7.2, the number of kernel threads that can be created per-process has been increased to 256. The maximum value for the MULTITHREAD system parameter has also been increased to 256.

4.13.2 New Method for Changing Kernel Thread Priorities

The SYS$SETPRI system service and the SET PROCESS/PRIORITY DCL command both take a process identification value (PID) as an input and therefore affect only a single kernel thread at a time. If you want to change the base priorities of all kernel threads in a process, a separate call to SYS$SETPRI or invocation of the SET PROCESS/PRIORITY command must be done for each thread.

In OpenVMS Alpha Version 7.2, a new value for the 'policy' parameter to the SYS$SETPRI system service has been added. If JPI$K_ALL_THREADS is specified, the call to SYS$SETPRI changes the base priorities of all kernel threads in the target process.

Also, the ALL_THREADS qualifier has been added to the SET PROCESS /PRIORITY DCL command which provides the same support.

4.13.3 Detecting Thread Stack Overflow

The default user stack in a process can expand on demand to be quite large, so single threaded applications do not typically run out of user stack. When an application is written using DECthreads, each thread gets its own user stack, which is a fixed size. If the application developer underestimates the stack requirements, the application may fail due to a thread overflowing its stack. This failure is typically reported as an access violation and is very difficult to diagnose. To address this problem, yellow stack zones have been introduced in OpenVMS Version 7.2 and are available to applications using DECthreads.

Yellow stack zones are a mechanism by which the stack overflow can be signaled back to the application. The application can then choose either to provide a stack overflow handler or do nothing. If the application does nothing, this mechanism helps pinpoint the failure for the application developer. Instead of an access violation being signaled, a stack overflow error is signaled.

4.14 New Features of OpenVMS Debugger

The OpenVMS Debugger contains the following new features:

- Windows client/server interface
- Heap Analyzer displays 64-bit space (Alpha only)
- Support for C++ Version 5.5 and later (Alpha only)
- PTHREAD Command

The following sections describe these OpenVMS Debugger enhancements. For more detailed information, see the *OpenVMS Debugger Manual*.

4.14.1 Client/Server Interface

The OpenVMS Debugger Version 7.2 features a new client/server interface that allows you to use Windows-based client applications to debug programs that reside and run on OpenVMS VAX or Alpha systems.

The OpenVMS server provides a fully cached DCE-RPC interface that can support up to 32 simultaneous client connections. These connections can occur across mixed desktop platforms such as DECwindows Motif, Windows NT, and Windows 95. Typically, a single client/server connection is maintained; however, additional parallel connections can be initiated to support team debugging, remote support, or classroom training.

Client applications are available for DECwindows Motif, Microsoft Windows NT, and Windows 95 systems. These clients are capable of running remote debug sessions with OpenVMS servers over TCP/IP, DECnet, or UDP network transports. Each client can support simultaneous connections to an unlimited number of OpenVMS servers on both VAX and Alpha platforms.

All Windows clients are implemented with Microsoft Visual C++/MFC and include additional features such as access to web-based HTML documentation and user-definable buttons for accessing other desktop tools (including Telnet sessions for managing the servers).

The new client/server interface provides new flexibility for debugging OpenVMS applications. You can run several sessions simultaneously from a PC, share debugging context between several users on different PCs, and swap back and forth between different debugging sessions.

The new client/server environment supports the following types of configurations:

- **One-to-One** (one server; one client)

  A typical debugging configuration.

- **One-to-Many** (one client; multiple servers)

  This allows the client to switch between different sessions to debug related or interacting programs simultaneously.

- **Many-to-One** (multiple clients; one server)

  This allows several users to collectively debug a single program, or to use the session as a teaching tool.

- **Many-to-Many** (multiple clients; multiple servers)

  This allows many users to team debug multiple programs.

These different configurations can consist of any combination of OpenVMS VAX and Alpha servers and Windows clients (Intel and Alpha). In addition, DECwindows Motif clients can simultaneously access debugging sessions on these configurations.

4.14.2 Heap Analyzer Displays 64-Bit Space (Alpha Only)

The Heap Analyzer has been enhanced to display 64-bit address space when required. The Heap Analyzer also displays 64-bit addresses where appropriate, but uses 32-bit addressing when that is sufficient.

The Zoom pulldown menu has two new entries to help scale the view of the entire memory space used by programs that take advantage of 64-bit address space. These new entries are:

Extremely Far
Extraordinarily Far

A new pulldown menu, the View menu, contains entries for the views listed in the Views window. Entries include:

All Views
Images
P0 Regions
P2 Region (if accessed by a 64-bit system routine in the program being analyzed)
Pxx Region (named virtual region created by $CREATE_REGION_64)

These new features of the Heap Analyzer makes it much easier to debug OpenVMS applications that use the 64-bit address space.

4.14.3 Support for C++ Version 5.5 and Later (Alpha Only)

OpenVMS Version 7.2 provides you with the ability to debug C++ programs with the OpenVMS Alpha debugger. Specifically, the OpenVMS Alpha debugger supports the following C++ features:

- C++ names and expressions, including:
  - Explicit and implicit **this** pointer to refer to class members
  - Scope resolution operator (::)
  - Member access operators: period (.) and right arrow (->)
  - Template instantiations

- Setting breakpoints in:
  - Member functions, including static and virtual functions
  - Overloaded functions
  - Constructors and destructors
  - Template instantiations
  - Operators
- Calling functions, including overloaded functions
- Debugging programs containing a mixture of C++ code and code in other languages

This new support enables you to use the OpenVMS debugger to debug OpenVMS Alpha applications written in DEC C++ Version 5.5 and later.

4.14.4 PTHREAD Command

When debugging programs that use DECthreads Version 3.13 or greater, you can directly access the DECthreads Debugger with the PTHREAD command.

The syntax of the PTHREAD command is

PTHREAD *command*

where *command* is a valid DECthreads Debugger command. The OpenVMS Debugger passes the command to the DECthreads Debugger for execution. You can get help on DECthreads debugger commands by typing `PTHREAD HELP`. For example:

```
DBG_1> PTHREAD HELP
    conditions [-afhwqrs] [-N <n>] [id]...: list condition variables
    exit: exit from DECthreads debugger
    help [topic]: display help information
    keys [-v] [-N <n>] [id]...: list keys
    mutexes [-afhilqrs] [-N <n>] [id]...: list mutexes
    quit: exit from DECthreads debugger
    show [-csuv]: show stuff
    squeue [-c <n>] [-fhq] [-t <t>] [a]: format queue
    stacks [-fs] [sp]...: list stacks
    system: show system information
    threads [-1] [-N <n>] [-abcdfhklmnor] [-s <v>] [-tz] [id]...: list threads
    tset [-chna] [-s <v>] <id>: set state of thread
    versions: display versions
    write <st>: write a string
All keywords may be abbreviated: if the abbreviation is ambiguous,
the first match will be used. For more help, type 'help <topic>'.
DBG_1>
```

The PTHREAD command gives you direct access to the DECthreads Debugger within the context of a debugging session. This can provide much more information to help debug the application.

4.15 OpenVMS RTL Library (LIB$)

This section describes the RTL routine LIB$GET_LOGICAL.

4.15.1 LIB$GET_LOGICAL

LIB$GET_LOGICAL provides a simplified interface to the $TRNLNM system service. It provides most of the features found in $TRNLNM with some additional benefits. For string arguments, all string classes supported by the Run-Time Library are understood. The list of item descriptors, which may be difficult to construct in high-level languages, is handled internally by LIB$GET_LOGICAL.

See the *OpenVMS RTL Library (LIB$) Manual* for more information about routine LIB$GET_LOGICAL.

4.16 Record Management Services (RMS) Enhancements

The following sections describe the RMS enhancements included with the OpenVMS Version 7.2 release.

4.16.1 Coordinated Universal Time (UTC) Support

With OpenVMS Version 7.2, both RMS and the File Access Listener (FAL) have been enhanced to support the 128-bit Coordinated Universal Time (UTC) format for the exchange of date and time information about files (such as file creation or file revision).

With this enhancement, RMS and FAL support two formats for the exchange of file date and time information:

- New UTC format. This enhancement takes advantage of the DECnet Data Access Protocol (DAP) option for transferring times using the 128-bit UTC format. As of this version, this has been made the default for OpenVMS systems.
- Current 2-digit year format. A 2-digit year is used in the following 18-byte format in accordance with the DAP specification: DD-MON-YYHH:MM:SS

In previous versions of RMS and FAL, as part of their own implementation (not part of the DAP specification), the 2-digit year field pivoted about the year 1970. YYs from 70 to 99 map to 1970 through 1999; YYs from 00 to 69 map to 2000 through 2069. Thus, this scheme presents no Year 2000 problem for previous versions in the immediate future.

If the client requesting the file transfer in the system configuration message has set the system capability (SYSCAP) indicating "support for binary date/time format," OpenVMS will use the UTC format by default. And as of this version, it will be set for any OpenVMS client.

If the UTC capability is not set by a non OpenVMS client, then the current 2-digit year scheme will remain in effect. Note that the pivot around the 1970 year, implemented in the OpenVMS RMS and FAL code, is not part of the DAP specification; therefore, it cannot be presumed that either a pivot or a pivot around the specific year 1970 is implemented by non OpenVMS clients.

4.16.2 Support for New File Length Hint Attribute (Alpha Only)

RMS is now capable of storing, retrieving, and maintaining a **file length hint**. A file length hint is a pair of quadword integer fields (16 bytes) as follows:

1. Record count (bytes 0-7): the number of data records written to the file using record I/O ($PUT).
2. User data byte count (bytes 8-15): the total number of user data bytes in the file (excluding any overhead bytes added by RMS).

For sequential files with a record format of variable (VAR) or variable with fixed control (VFC) on an On-Disk Structure Level 5 (ODS-5) volume, RMS will maintain the file hint, provided:

- The file is written only using unshared RMS record I/O.
- The file does not have journaling enabled.

The XAB$_FILE_LENGTH_HINT item code may be used with an item list XAB on $OPEN or $DISPLAY operations to sense the file length hint values. A SETMODE may be used with a $CLOSE operation to set the file length hint counts. The SETMODE will override any counts that RMS may be concurrently maintaining.

The XAB$_FILE_LENGTH_HINT XABITM requires a 16-byte buffer for the two quadwords. These fields are maintained as a set: either both fields are valid or invalid.

The most significant (sign) bit of each quadword is used to indicate whether the associated count is valid. A sequential file with VAR or VFC format that is created on an ODS-5 volume, had any data added to it using RMS record I/O ($PUTs), and has met the conditions indicated above should have valid counts. If, however, at some point in time, some data are written to a file using RMS block I/O, for example, then the sign bits will be set on file deaccess to indicate the counts are invalid. The last count maintained in each field is retained as a hint of what its last valid value was, but the sign bit being set indicates it is stale.

If these fields have never been modified by RMS for a file on an ODS-5 volume, then the contents of each quadword will be 8 bytes of 0xFF. For example, after a file originally created and maintained on an On-Disk Structure Level 2 (ODS-2) volume is converted from ODS-2 format to ODS-5 format, these fields will contain 8 bytes of 0xFF.

The counts in these fields are invalidated if a truncate-on-put is done, except if the truncate is to zero.

If a SENSEMODE using this item code is requested for a non-ODS-5 file, the contents returned for each quadword will be 8 bytes of 0xFF. A SETMODE using this item code for a non-ODS-5 file will be ignored.

The file length hint is not supported for DECnet operations; it is ignored. If a SENSE is attempted, 8 bytes of 0xFF will be returned to the user buffer for each quadword.

4.16.3 New Qualifier for the Analyze/RMS_File Utility

With OpenVMS Version 7.2, the Analyze/RMS_File utility has a new qualifier /UPDATE_HEADER. The /UPDATE_HEADER qualifier attempts to update the following attributes in the header of the file: longest record length (LRL) and/or file length hint attribute. You must use this qualifier in combination with either /STATISTICS or /CHECK (the default). This qualifier only applies to sequential file organizations and is ignored for any other file organization.

For more detailed information, see the *OpenVMS Record Management Utilities Reference Manual*.

4.16.4 File User Characteristics Item for XAB (XABITM)

With OpenVMS Version 7.2, the file user characteristic item XAB$_UCHAR_PRESHELVED is available for use with the XAB (XABITM) item list of RMS. The XAB$_UCHAR_PRESHELVED item indicates that a file is shelved but also kept online.

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

4.16.5 RMS Global Buffer Performance Enhancements

OpenVMS Version 7.2 includes two enhancements that improve RMS global buffer performance. The two enhancements are as follows:

- RMS implements a new algorithm for global buffer management that dramatically improves scalability. The performance associated with the previous algorithm effectively limited the maximum number of global buffers on large, shared files. With this change, you may increase the number of global buffers on these files to the full limit of 32,767 to fully exploit large memory systems.
- RMS now synchronizes access to the global section that is used for RMS global buffers by using inline atomic instruction sequences rather than distributive locking. This change allows more concurrent access to the section, particularly on symmetric multiprocessing machines (SMP).

The first change benefits those who wish to use very large global buffer counts. The second change helps any application using global buffers where contention on the global section itself is a bottleneck.

Note

By increasing the number of global buffers on specific files, you may need to increase in size some of the system resources. In particular, the sysgen parameters GBLPAGES, GBLPAGFILE, or GBLSECTIONS may need to be increased. In addition, you may need to increase the process working set size and the page file quota.

4.17 Scheduling Processes with Soft Affinity (Alpha only)

In OpenVMS Alpha Version 7.2, the scheduling algorithm for soft affinity has been improved. **Soft affinity** allows a kernel thread to be scheduled on the last processor on which it has run. This capability preserves its investment in the cache and translation buffers.

The new algorithm bases its decision on whether a soft affinity process should execute on the current CPU by collecting and using process scheduling data. The system collects more information about how processes are performing on each CPU by collecting the following data at each process priority level:

- Process runtime
- Process waittime (how long it waited on the COM queue before it was selected to run)
- Interrupt time incurred by the process

Soft affinity can be enabled in two ways:

- The $SET_IMPLICIT_AFFINITY system service enables soft affinity for a particular process or as the default for all future processes.

  The CAP$V_IMPLICIT_ALL_THREADS flag enables soft affinity for all kernel threads in a process.

- The new DCL command SET PROCESS/AFF=SOFT enables soft affinity for a process.

Prior to OpenVMS Alpha Version 7.2, the DCL command SET PROCESS/AFF /SET=1 enabled hard (or explicit) affinity for a process, but there was no DCL command that enabled soft affinity. As of OpenVMS Alpha Version 7.2, the qualifier /AFF has *hard* and *soft* values associated with it. If neither is specified, the default is *hard*, which preserves the previous behavior.

In OpenVMS Alpha Version 7.2, the DCL command SHOW PROCESS/ALL displays the soft affinity setting. This command displays the following new line:

```
Soft affinity: on
```

or if soft affinity is not enabled:

```
Soft affinity: off.
```

Two system parameters are associated with soft affinity:

- AFFINITY_SKIP

  In OpenVMS Alpha Version 7.2, AFFINITY_SKIP is obsolete.

- AFFINITY_TIME

  As of OpenVMS Alpha Version 7.2, AFFINITY_TIME specifies how long (in seconds) the number of voluntary waits a process has incurred should be counted before determining if the process is I/O bound or compute bound. The default value is 3 seconds.

4.18 Security Features (Alpha Only)

This section contains information about security changes in OpenVMS Version 7.2.

4.18.1 Per Thread Security

This section discusses per-thread security, a new feature in OpenVMS Alpha Version 7.2.

Per-thread security permits each thread of execution within a multithreaded process to have an individual security profile. In OpenVMS Version 7.2, the impersonation system services and underlying system framework have been enhanced to support per-thread security profiles.

Security profile information previously contained in various process level data structures and data cells is now stored in a single data structure, the Persona Security Block (PSB), which is then bound to a thread of execution. All associated references within OpenVMS have been redirected accordingly. Every process in the system has at least one PSB that is the **natural persona** of the process. The natural persona is created during process creation.

Interaction between a thread manager (for example, the thread manager incorporated within DECthreads) and the security subsystem provides for the automatic switching of profiles as threads are scheduled for execution.

Refer to Appendix B, "Considerations for OpenVMS Systems" in the *Guide to DECthreads* for further information about threading capabilities available on OpenVMS systems.

4.18.1.1 Benefits

The primary consumer of per-thread security is a multithreaded server with threads that impersonate clients. These threads appear to the system as the clients in regard to audits, access checks, rights processing, and so on. This is a benefit to those writing a system level server application that processes requests on behalf of users. These applications can be coded using the DECthreads thread model and the system's built-in impersonation services to have the system automatically perform the security checking on behalf of the requesting client.

When kernel threads were implemented in the OpenVMS operating system, modifications to one thread's security information (privileges, rights, and identity information) could be inadvertently passed to another thread if the threads are scheduled on different processors simultaneously, as kernel threads are designed to do.

Per-thread security ensures that this security information is handled properly. Each user thread in a process has a fully separate security profile. When the user thread is scheduled, the security profile for that thread is automatically switched as well.

Refer to the *OpenVMS Guide to System Security* for more information about per-thread security.

4.18.2 Enhanced Persona Support

Previous versions of OpenVMS included system services that allowed a privileged OpenVMS process to create and use personae. These system services are as follows:

- $PERSONA_ASSUME
- $PERSONA_CREATE
- $PERSONA_DELETE

Enhancements and additions to these services are a new feature in OpenVMS Alpha Version 7.2. These enhancements support subject security credentials other than OpenVMS. Initially, Windows NT-style security credentials are supported.

4.18.2.1 New Persona System Services

The following new system services support persona lookup and retrieval and persona extensions. These system services are documented in *OpenVMS System Services Reference Manual: GETQUI–Z*. Refer to the *OpenVMS Guide to System Security* for additional information about these services.

- $PERSONA_CLONE
- $PERSONA_CREATE_EXTENSION
- $PERSONA_DELETE_EXTENSION
- $PERSONA_DELEGATE
- $PERSONA_EXTENSION_LOOKUP
- $PERSONA_FIND
- $PERSONA_MODIFY

- $PERSONA_QUERY
- $PERSONA_RESERVE

Note

The persona extension services ($PERSONA_CREATE_EXTENSION, $PERSONA_DELETE_EXTENSION, $PERSONA_DELEGATE, $PERSONA_EXTENSION_LOOKUP, and $PERSONA_RESERVE) are under development and are subject to change in a release following OpenVMS Version 7.2.

4.19 Shared Address Data (Alpha Only)

Using shared address data on OpenVMS Alpha systems improves performance at the following times:

- At run time, shared address data saves physical memory because of increased memory sharing between processes.
- At image activation, shared address data reduces CPU and I/O time because fixup is performed at installation time.

Related Terms

Explanations of terms related to shared address data follow.

- Image section

  An image consists of a number of image sections. An image section can contain:

  - Instructions (code)
  - Read-only data (constants)
  - Read/write data

- Shared known images

  You can make an image a shared known image by using the following command:

  ```
  $ INSTALL ADD image-name /SHARED
  ```

  When you enter this command, the Install utility creates global sections for read-only image sections, allowing the sections to be shared by all the processes that run the image.

- Address data

  One kind of image section contains address data. At execution time, address data sections are read-only. However, the addresses are not known until image activation; therefore, the image section is read/write until the end of image activation. Addresses for a shareable image generally vary with the process, because different processes are likely to have different collections of mapped images.

- Shared address data

The shared address data feature assigns unique P1 space addresses for shareable images from the P1 image region. (The IMGREG_PAGES system parameter determines the size of the P1 space.) With the assigned address, the Install utility determines the content of an address data section when the image is installed. A global section is created to allow shared access to each address data image section.

Executable (main) images can also use shared address data sections; these images are not assigned P1 addresses, however, because the base address for an executable image is determined when the image is linked.

4.19.1 Creating Shared Address Data Sections

To install an image with shared address data, you use the new keyword ADDRESS_DATA with the INSTALL/SHARED command:

```
$ INSTALL CREATE/SHARED=ADDRESS_DATA imagename
```

Note that when you install an image (Image A, for example) with shared address data, and Image B is called by Image A, Image B must also be installed with shared address data. Two corollaries of this rule are the following:

- If Image B is not installed with shared address data, the Install utility displays a message indicating this, and Image A is installed *without* shared address data.
- If Image B is removed (with the command $ INSTALL REMOVE B), when the image activator subsequently activates Image A, the activator will create *unshared* address data sections for Image A.

4.19.2 New INSTALL Command Keywords

On OpenVMS Alpha systems, you can use the [NO]ADDRESS_DATA keyword with the following commands:

- INSTALL> CREATE/SHARED[=[NO]ADDRESS_DATA]
- INSTALL> REPLACE/SHARED[=[NO]ADDRESS_DATA]

When you use the ADDRESS_DATA keyword with either command, P1 space addresses are assigned for shareable images. With the assigned addresses, the Install utility can determine the content of an address data section when the image is installed rather than when it is activated, reducing CPU and I/O time. A global section is created to allow shared access to address data image sections.

Example:

```
INSTALL> CREATE/SHARED=ADDRESS_DATA WRKD$:[MAIN]INFOSHR
```

The command in this example installs the INFOSHR file as a shared known image and creates shared global sections for code sections and read-only data sections. Because the command includes the ADDRESS_DATA keyword, address data is also created as a shared global section.

4.20 System Dump Analyzer (SDA) Enhancements (Alpha Only)

The following sections describe the new SDA features included in OpenVMS Version 7.2.

4.20.1 New and Enhanced Commands

The following SDA CLUE extension commands, SDA extension routines, and SDA commands are either new or have new or changed qualifiers or parameters for OpenVMS Alpha Version 7.2. For further information about these commands, see the *OpenVMS Alpha System Analysis Tools Manual*.

4.20.1.1 CLUE CALL_FRAME Extension Command

The following table shows the qualifiers for the CLUE CALL_FRAME extension command:

Qualifier	Meaning
/ADDRESS=*n*	Specifies the PCB address of the desired process when used with CLUE CALL_FRAME/PROCESS.
/CPU [cpu-id | ALL	Indicates that the call frame for a CPU is required. The CPU should be specified by its number or by using ALL to indicate all CPUs.
/IDENTIFICATION=*n*	Specifies the identification of the desired process when used with CLUE CALL_FRAME/PROCESS.
/INDEX=*n*	Specifies the index of the desired process when used with CLUE CALL_FRAME/PROCESS.
/PROCESS [process-name | ALL]	Indicates that the call frame for a process is required. The process should be specified with one of the qualifiers /ADDRESS, /IDENTIFICATION, or /INDEX; or by its name, or by using ALL to indicate all processes.

4.20.1.2 CLUE ERRLOG Extension Command:/OLD qualifier

The CLUE ERRLOG extension command has a new qualifier, /OLD. The /OLD qualifier dumps the errorlog buffers into a file using the old errorlog format. The default action, if /OLD is not specified, is to dump the errorlog buffers in the common event header format.

4.20.1.3 CLUE FRU Extension Command

The CLUE FRU extension command outputs the Field Replacement Unit (FRU) table to a file for display by DECevent.

4.20.1.4 CLUE REGISTER Extension Command

The Alpha SDA new command CLUE REGISTER displays the active registers for the crash CPU. The CLUE REGISTER command is only valid when analyzing crash dumps.

4.20.1.5 CLUE SG Extension Command: /CRAB qualifier

The CLUE SG extension command displays the scatter-gather map. The /CRAB qualifier displays the ringbuffer for the specified Counted Resource Allocation Block (CRAB). The default action is to display the ringbuffer for all CRABs.

4.20.1.6 CLUE SYSTEM/LOGICAL Extension Command

The Alpha SDA new command CLUE SYSTEM/LOGICAL command displays the contents of the shared logical name tables in the system.

4.20.1.7 SDA$GET_CURRENT_CPU Extension Routine

The SDA$GET_CURRENT_CPU extension routine gets the CPU database address of the currently selected CPU.

4.20.1.8 SDA$SET_CPU Extension Routine

The SDA$SET_CPU extension routine sets a new SDA CPU context.

4.20.1.9 SDA$SET_PROCESS Extension Routine

The SDA$SET_PROCESS extension routine sets a new SDA process context.

4.20.1.10 SHOW DUMP Command: /MEMORY_MAP Qualifier

Displays the memory map of a full dump.

4.20.1.11 SHOW EXECUTIVE Command: EXECLET-NAME Parameter

The Alpha SDA command SHOW EXECUTIVE has a new parameter, **execlet-name**. If **execlet-name** is given, SDA only displays the data for the specified execlet. Wildcards can be used in **execlet-name**, in which case SDA displays data for all matching execlets.

The benefit of the **execlet-name** parameter is to reduce the volume of output necessary to obtain the desired information.

4.20.1.12 SHOW EXECUTIVE Command: /SUMMARY Qualifier

The Alpha SDA command SHOW EXECUTIVE has a new qualifier. The /SUMMARY qualifier displays a single line of output for each loadable image.

4.20.1.13 SHOW GSD Command: New Qualifiers

The SHOW GSD command has the following new qualifiers:

Qualifier	Meaning
/GLXGRP	Displays information in the group global section descriptors of a Galaxy system
/GLXSYS	Displays information in the system global section descriptors of a Galaxy system

4.20.1.14 SHOW LOCK Command: New Qualifiers

The SHOW LOCK command has the following new qualifiers:

Qualifier	Meaning
/BLOCKING	Displays only the locks that have a blocking AST specified or attached
/BRIEF	Displays a brief one line of lock information
/CONVERT	Displays only the locks that are on the conversion queue
/GRANTED	Displays only the locks that are on the granted queue
/POOL	Displays the lock managers poolzone information, which contains the lock blocks (LKB) and resource blocks (RSB)
/STATUS=(*keyword* [,*keyword...]*)	Displays only the locks that have the specified status bits set in the LKB$L_STATUS field. For the keywords and their meanings for this qualifier, see the *OpenVMS Alpha System Analysis Tools Manual*.
/SUMMARY	Displays summary data and performance counters

Qualifier	Meaning
/WAITING	Displays only the waiting locks

4.20.1.15 SHOW PAGE_TABLE Command: Qualifiers

The Alpha SDA command SHOW PAGE_TABLE has the following new qualifiers:

Qualifier	Meaning
/INVALID_PFN [=*option*]	This qualifier, which is valid on platforms that supply an I/O memory map, causes SDA to display only page table entries that map to PFNs that are not in the system's private memory, nor in Galaxy shared memory, nor are I/O access pages.
/NONMEMORY_PFN [=*option*]	This qualifier, supported on all platforms, causes SDA to display only page table entries that are neither in the system's private memory nor in Galaxy shared memory.
/PTE_ADDRESS	Specifies that the range given is of PTE addresses instead of the virtual addresses mapped by the PTEs.
/SECTION_INDEX=*n*	Displays the page table for the range of pages in the global section or pageable part of a loaded image.

Note

Both /INVALID_PFN and /NONMEMORY_PFN qualifiers allow two optional keywords, READONLY and WRITABLE. If neither keyword is given, all relevant pages are displayed. If READONLY is given, only pages marked for no write access are displayed. If WRITABLE is given, only pages that allow write access are displayed. For example, SHOW PAGE_TABLE=ALL/INVALID_PFN=WRITABLE would display all system pages whose protection allows write, but which map to PFNs that do not belong to this system.

4.20.1.16 SHOW PAGE_TABLE/FREE Command: /HEADER Qualifier

The Alpha SDA command SHOW PAGE_TABLE /FREE has a new qualifier. The /HEADER=address qualifier causes SDA to display the free list for the specified private page table.

4.20.1.17 SHOW PARAMETER Command

The Alpha SDA new command SHOW PARAMETER displays the name, location, and value of one or more SYSGEN parameters at the time that the system dump is taken.

The SHOW PARAMETER command has the following parameter:

Parameter	Meaning
SYSGEN_parameter	The name of a parameter to be displayed. The name given may include wildcards. However, a truncated name is not recognized, unlike the equivalent SYSGEN and SYSMAN commands.

The SHOW PARAMETER has the following qualifiers:

Qualifier	Meaning
/ACP	Displays all Files–11 parameters.
/ALL	Displays the values of all parameters except the special control parameters.
/CLUSTER	Displays all parameters specific to clusters.
/DYNAMIC	Displays all parameters that can be changed on a running system.
/GALAXY	Displays all parameters specific to Galaxy systems.
/GEN	Displays all general parameters.
/JOB	Displays all Job Controller parameters.
/LGI	Displays all LOGIN security control parameters.
/MAJOR	Displays the most important parameters.
/MULTIPROCESSING	Displays parameters specific to multiprocessing.
/PQL	Displays the parameters for all default process quotas.
/RMS	Displays all parameters specific to OpenVMS Record Management Services (RMS).
/SCS	Displays all parameters specific to OpenVMS Cluster System Communications Services.
/SPECIAL	Displays all special control parameters.
/STARTUP	Displays the name of the site-independent startup procedure.
/SYS	Displays all active system parameters.
/TTY	Displays all parameters for terminal drivers.

4.20.1.18 SHOW PFN_DATA Command: /UNTESTED and /PRIVATE [=address] Qualifiers

The Alpha SDA command SHOW PFN_DATA has two new qualifiers: /PRIVATE [=address] and /UNTESTED. The /PRIVATE [=address] qualifier causes SDA to display private PFN lists. If no address is given, all private PFN lists are displayed; if an address is given, only the PFN list whose head is at the given address is displayed. The /UNTESTED qualifier causes SDA to display the state of the untested PFN list that was set up for deferred memory testing.

4.20.1.19 SHOW PROCESS Command: New Qualifiers

The Alpha SDA command SHOW PROCESS has the following new qualifiers:

Qualifier	Meaning
/BRIEF	This qualifier, when used with the /LOCKS qualifier, causes SDA to display each lock owned by the current process in brief format; that is, one line for each lock.
/CHANNEL [=PID_ONLY]	Displays information about the I/O channels assigned to the process. If the keyword FID_ONLY is given, SDA does not attempt to translate the FID (File ID) to a file name when invoked with ANALYZE/SYSTEM.
/FANDLE	Displays the data on the process's fast I/O handles.
/INVALID_PFN [=*option*]	This qualifier, which is valid on platforms that supply an I/O memory map, causes SDA to display only page table entries that map to PFNs that are not in the system's private memory, nor in Galaxy shared memory, nor are I/O access pages.
/LOCKS [/BRIEF]	Displays the lock management locks owned by the current process. The /LOCKS [/BRIEF] qualifier produces a display similar in format to that produced by the SHOW LOCK command. See also the /BRIEF qualifier description.
/NONMEMORY_PFN [=*option*]	This qualifier, supported on all platforms, causes SDA to display only page table entries that are neither in the system's private memory nor in Galaxy shared memory.
/PERSONA [=address]	Displays all persona security blocks (PSBs) held in the PERSONA ARRAY of the process, and then lists selected information contained in each initially listed PSB. The selected information includes the contents of the following cells inside the PSB: Flags Reference count Execution mode Audit status Account name UIC Privileges Rights enabled mask If a PSB address is specified, the above information is provided for that specific PSB only.
/PERSONA/RIGHTS	Displays all the /PERSONA [=address] information and additional selected information, including all the Rights and their attributes currently held and active for each persona security block (PSB).
/PERSONA/RIGHTS /AUTHORIZED	Displays all the /PERSONA [=address] information and additional selected information, including all the Rights and their attributes authorized for each persona security block (PSB).

4.20.1.20 SHOW PROCESS/PAGE_TABLES Command

The SHOW PROCESS/PAGE_TABLES command has the following new qualifiers:

Qualifier	Meaning
/PTE_ADDRESS	Specifies a PTE address instead of the virtual address mapped by the PTE.
/SECTION_INDEX=*n*	Displays the page table for the range of pages in the global section or pageable part of a loaded image, or the process section specified when used with the /PAGE_TABLE qualifier. Except for global sections where global page table entries (GPTEs) are displayed, one of the qualifiers /L1, /L2, or /L3 can also be specified. Displays the PST for the single process section specified when used with the /PROCESS_SECTION_TABLE qualifier. The /SECTION_INDEX=*n* qualifier is ignored if neither the /PAGE_TABLE nor the /PROCESS_SECTION_TABLE qualifiers is specified.
/INVALID_PFN [=*option*]	This qualifier, which is valid on platforms that supply an I/O memory map, causes SDA to display only page table entries that map to PFNs that are not in the system's private memory, nor in Galaxy shared memory, nor are I/O access pages.
/NONMEMORY_PFN [=*option*]	This qualifier, supported on all platforms, causes SDA to display only page table entries that are neither in the system's private memory nor in Galaxy shared memory.

Note

Both /INVALID_PFN and /NONMEMORY_PFN qualifiers allow two optional keywords, READONLY and WRITABLE. If neither keyword is given, all relevant pages are displayed. If READONLY is given, only pages marked for no write access are displayed. If WRITABLE is given, only pages that allow write access are displayed. For example, SHOW PROCESS ALL/PAGE_TABLE=ALL/INVALID_PFN=WRITABLE would display all process pages (for all processes) whose protection allows write, but which map to PFNs that do not belong to this system.

4.20.1.21 SHOW RESOURCE Command: New Qualifiers

The SHOW RESOURCE command has the following new qualifiers:

Qualifier	Meaning
/BRIEF	Displays a brief one line of the resource information
/CONTENTION [=ALL]	Displays only resources that have at least one lock on either the waiting or conversion queue
/STATUS=(*keyword* [*keyword*,])	Displays only resources that have the specified status bits set in the RSB$L_STATUS field. For the keywords and their meanings for this qualifier, see the *OpenVMS Alpha System Analysis Tools Manual*.

4.20.1.22 SHOW STACK Command: /PHYSICAL Qualifier

The Alpha SDA command SHOW STACK has a new qualifier: /PHYSICAL. The /PHYSICAL qualifier causes SDA to treat the start and/or end address in the given range as a physical address. This qualifier is only relevant when a range is specified.

4.20.1.23 SHOW STACK Command: RANGE Parameter

The Alpha SDA command SHOW STACK has a new parameter, **range**. The **range** parameter displays a range of memory locations in stack format. You can express a **range** using the following syntax:

Value	Meaning
m:n	Range of addresses from *m* to *n*
m;n	Range of addresses starting at *m* and continuing for *n* bytes

4.20.1.24 SHOW SUMMARY Command: /USER=username Qualifier

The Alpha SDA command SHOW SUMMARY has a new qualifier /USER=username. If /USER=username is given, SDA only displays the processes of the specified user. Wildcards can be used in /USER=username, in which case SDA displays processes of all matching users.

The benefit of the /USER=username qualifier is to reduce the volume of output necessary to obtain the desired information.

4.20.1.25 SHOW SYMBOL/ALL Command: /ALPHA and /VALUE Qualifiers

The Alpha SDA command SHOW SYMBOL /ALL has two new qualifiers. The /ALPHA qualifier causes SDA to display the symbols sorted only in alphabetical order. The /VALUE qualifier causes SDA to display the symbols sorted only in value order. The default is to display the symbols twice, sorted alphabetically and then by value.

4.20.1.26 SHOW SYMBOL Command: Enhanced to Allow Wildcards

The Alpha SDA command SHOW SYMBOL without the /ALL qualifier now causes SDA to allow standard wildcards in the **symbol-name** parameter. By default, matching symbols are displayed only in alphabetical order. If SHOW SYMBOL /VALUE is specified, then matching symbols are output sorted by value. If both /ALPHA and /VALUE are specified, matching symbols are displayed twice, sorted alphabetically and then by value.

The benefit of this new functionality is increasing the flexibility of the SHOW SYMBOL command for locating symbols whose name is only partially known.

4.20.1.27 VALIDATE PFN_LIST Command: /UNTESTED and /PRIVATE Qualifiers

The Alpha SDA command VALIDATE PFN_LIST has two new qualifiers: /PRIVATE and /UNTESTED. The /PRIVATE qualifier causes SDA to validate all private PFN lists. The /UNTESTED qualifier causes SDA to validate the state of the untested PFN list that was set up for deferred memory testing.

4.20.1.28 VALIDATE QUEUE Command: /PHYSICAL and /BACKLINK Qualifiers

The Alpha SDA command VALIDATE QUEUE has two new qualifiers: /PHYSICAL and /BACKLINK. The /PHYSICAL qualifier causes SDA to allow validation of queues whose header and links are physical addresses. The /BACKLINK qualifier causes SDA to allow doubly linked lists to be validated from the tail of the queue.

The benefit of the /BACKLINK qualifier is that if the queue is found to be broken when validated from the head of the queue, this qualifier can be used to narrow the list of corrupted entries.

4.20.2 SDA CLUE: DOSD File Analysis Capability

OpenVMS Alpha Version 7.2 allows you to write the system dump file to a device other than the system disk. With this enhancement, CLUE HISTORY information can be collected even when dump off system disk (DOSD) is in use. For SDA CLUE to locate the dump file to be analyzed, you need to perform the following steps:

1. Modify the command procedure SYS$MANAGER:SYLOGICALS.COM to add the system logical name CLUE$DOSD_DEVICE to point to the device where the dump file resides. You need to supply only the physical or logical device name without a file specification.
2. Modify the command procedure SYS$MANAGER:SYLOGICALS.COM to mount systemwide the device where the dump file resides. Otherwise, SDA CLUE cannot access and analyze the dump file.

In the following example, the dump file has been placed on device 3DUA25, which has the label DMP$DEV. You need to add the following commands to SYS$MANAGER:SYLOGICALS.COM:

```
$ MOUNT/SYSTEM/NOASSIST $3$DUA25: DMP$DEV DMP$DEV
$ DEFINE/SYSTEM CLUE$DOSD_DEVICE DMP$DEV
```

4.20.3 SDA Extensions

OpenVMS Alpha Version 7.2 includes extension routines that allow you to create additional System Dump Analyzer (SDA) commands for interpreting system dumps. You can now use a documented and supported interface to write custom SDA extensions tailored to your business needs. For a description of the the SDA extension routines, as well as how to create and debug an extension, see the *OpenVMS Alpha System Analysis Tools Manual*.

4.21 System Services

Several new system services have been added, and several existing services have been modified for OpenVMS Version 7.2. This section contains the new and changed services.

4.21.1 Support for New Service, $GETENV

The new system service, $GETENV, returns the value(s) of the specified console environment variable(s).

4.21.2 Support for Intra-Cluster Communication

New services have been created to support Intra-Cluster Communication, an applications programming interface (API) for local and clusterwide interprocess communications.

For more information about these services and their application, see Section 4.11 in this manual.

4.21.3 Support for Clusterwide Logicals

This section contains information about services that have been modified to support clusterwide logical names. The following table details the changes.

System Service	New Features
$CRELNM	Support added for clusterwide logicals. • Modified item, TABNAM • Modified item code, LNM$_STRING • Added condition value, SS$_INSFMEM
$CRELNT	Support added for clusterwide logicals. • Modified attribute, LNM$M_CREATE_IF • Modified arguments, PARTAB and TABNAM • Added condition value, SS$_INSFMEM • Modified condition value, SS$_IVLOGTAB • Modified Required Access or Privileges section
$DELLNM	Support added for clusterwide logicals. • Added condition value, SS$_INSFMEM • Modified Required Access or Privileges section
$GETSYI	Support added for clusterwide logicals. • Modified 64-bit descriptor field, BUFFER LENGTH • Added item codes, SYI$_CWLOGICALS and SYI$_SYSTYPE • All item codes work for clusterwide logicals
$TRNLNM	Support added for clusterwide logicals. • Modified arguments, ATTR and TABNAM • Added item code attributes, LNM$M_CLUSTERWIDE and LNM$M_INTERLOCKED • Modified Description section

For more information about these services, refer to the *OpenVMS System Services Reference Manual.*

4.21.4 Support for Extended File Specifications

This section contains information about services that have been modified to support Extended File Specifications. The following table details the changes.

System Service	New Features
$CREPRC	Support added for Extended File Specifications. • Modified service to include new symbolic name for the stsflg parameter.
$CVT_FILENAME	Support added for Extended File Specifications. • New service that converts a string from RMS format to file-system (ACP/QIO) format, or from file-system (ACP/QIO) format to RMS format.
$GETJPI	Support added for Extended File Specifications. • Modified service to include new item codes, JPI$_PARSE_STYLE_PERM, and JPI$_PARSE_STYLE_IMAGE.
$SETDDIR	Support added for Extended File Specifications. • Modified the Description section
$SET_PROCESS_PROPERTIESW	Support added for Extended File Specifications. • New service that sets a simple value associated with a process

For more information about these services, refer to the *OpenVMS System Services Reference Manual*.

4.21.5 Support for Fast Path

Two new services have been created to support Fast Path, $IO_FASTPATH and $IO_FASTPATHW.

For more information about Fast Path, refer to the *OpenVMS System Services Reference Manual: GETQUI–Z*, to the *OpenVMS I/O User's Reference Manual*, and see Table 2–1 in this manual.

4.21.6 Support for Personas

Previous versions of OpenVMS included system services that allowed a privileged OpenVMS process to create and use personae. Beginning with OpenVMS Version 7.2, several new system services are available to support persona lookup and retrieval and persona extensions.

For more information about these services and their application, see Section 4.18.2 in this manual and refer to the *OpenVMS System Services Reference Manual*.

5

Compaq Galaxy Software Architecture on OpenVMS Alpha

OpenVMS Alpha Version 7.2 introduces a model of computing that allows multiple instances of OpenVMS to execute cooperatively in a single computer. With Adaptive Partitioned Multiprocessing (APMP), many processors (and other physical resources) are partitioned in order to run multiple instances of operating systems. Each instance has assigned CPUs, memory, and I/O. The instances share a part of memory, and CPUs can be reassigned from one instance to another while the system runs. This computing environment can be dynamically adapted to changing application needs and workload demands.

Compaq's first implementation of the APMP model of computing is the Galaxy Software Architecture on OpenVMS Alpha. An evolution in OpenVMS functionality, OpenVMS Galaxy leverages proven OpenVMS cluster, symmetric multiprocessing, memory management, memory channel community management, and performance features to provide extremely flexible operational computing capabilities.

For companies looking to improve their ability to manage unpredictable, variable, or growing IT workloads, OpenVMS Galaxy technology provides a flexible way to dynamically reconfigure and manage system resources.

By running multiple instances of OpenVMS in a single computer, an OpenVMS Galaxy computing environment gives you quantum improvements in:

- Compatibility—Existing applications run without changes.
- Availability—Presents opportunities to upgrade software and expand system capacity without downtime.
- Scalability—Offers scaling alternatives that improve performance of SMP and cluster environments.
- Adaptability—Physical resources can be dynamically reassigned to meet changing workload demands.
- Cost of ownership—Fewer computer systems reduce system management requirements, floor space, and more.

An OpenVMS Galaxy computing environment is ideal for high-availability applications, such as:

- Database servers
- Transaction processing systems
- Data Warehousing
- Data Mining
- Internet servers

5.1 OpenVMS Galaxy Version 7.2 Features

With OpenVMS Alpha Version 7.2, you can create an OpenVMS Galaxy computing environment that allows you to:

- Run three instances of OpenVMS on an AlphaServer 8400
- Run two instances of OpenVMS on an AlphaServer 8200
- Run two instances of OpenVMS on an AlphaServer 4100
- Reassign CPUs between instances
- Perform independent booting and shutdown of instances
- Use shared memory for inter-instance communication
- Cluster instances within an OpenVMS Galaxy using the shared memory cluster interconnect
- Cluster instances with non-Galaxy systems
- Create applications using OpenVMS Galaxy APIs for resource management, event notification, locking for synchronization, and shared memory for global sections
- Use the Galaxy Configuration Utility to view and control the OpenVMS Galaxy environment
- Run a single-instance OpenVMS Galaxy on any Alpha system for application development

5.2 Getting More Information

For more information about how to create, manage, and use an OpenVMS Galaxy computing environment, refer to the *OpenVMS Alpha Galaxy Guide.* This book includes:

- OpenVMS Galaxy licensing, hardware, and configuration requirements
- Procedures for creating OpenVMS Galaxy computing environments on OpenVMS Alpha Server 8400, 8200, and 4100 systems
- Complete details about how to use all of the OpenVMS Galaxy features and capabilities available in OpenVMS Alpha Version 7.2

Note that release notes about the Galaxy Software Architecture on OpenVMS are included in the *OpenVMS Alpha Galaxy Guide.*

A

New OpenVMS System Messages

This release includes new messages for the following OpenVMS facilities:

- ACME, Authentication and Credential Management
- ANALDISK, Analyze/Disk_Structure Utility
- BACKUP, Backup Utility
- Cluster Port Driver
- INSTALL, Install Utility
- IOGEN
- MOUNT, Mount Utility
- OPCOM, Operator Communication Process
- PCSI, POLYCENTER Software Installation Utility
- PCSIUI, POLYCENTER Software Installation Utility
- POWER, AlphaServer 8200 Series Power Monitoring
- RMS, OpenVMS Record Management Services
- SDA, System Dump Analyzer
- SET, SET Command and SET Utility
- SET, SET PASSWORD Command
- SYSINIT, System Initialization
- SYSTEM, System Services
- UAF, Authorize Utility

The Help Message database has been updated with all the new messages. This appendix alphabetically lists and describes only those messages that might occur before the system is fully functional; that is, when the Help Message database is not accessible.

For information about the HELP/MESSAGE command and qualifiers, see DCL help (type HELP HELP/MESSAGE at the DCL prompt) or refer to *OpenVMS System Messages: Companion Guide for Help Message Users*.

ACTERR, error activating ICBM

Facility: IOGEN

Explanation: OpenVMS was unable to activate the proper IOGEN Configuration Building Module during device autoconfiguration. Most likely, the image SYS$SHARE:SYS$ICBM.EXE is not present on your system disk or is not installed.

User Action: Check to see whether the image file exists and verify that it is installed as a known image. (For instructions on installing a known image, refer to the Install utility chapter in the *OpenVMS System Management Utilities Reference Manual*.) If you still have problems, reinstall OpenVMS onto your system disk or contact a Compaq support representative for assistance.

ALHOMERR, I/O error reading alternate home block on 'device-name'

Facility: MOUNT, Mount Utility

Explanation: An I/O error occurred while attempting to read the alternate home block on the volume being mounted. An accompanying message provides additional information about the error. This is a warning message; the mount is allowed to proceed.

User Action: To avoid future errors, repair the named volume using the DCL command ANALYZE/DISK_STRUCTURE/REPAIR.

ALHOMINV, invalid alternate home block on 'device-name'

Facility: MOUNT, Mount Utility

Explanation: The alternate home block on the volume being mounted does not have a valid format. This is a warning message; the mount is allowed to proceed.

User Action: To avoid future errors, repair the named volume using the DCL command ANALYZE/DISK_STRUCTURE/REPAIR.

CI port Virtual Circuit SETCKT OPEN timeout: Restart Port

Facility: Cluster Port Driver

Explanation: When the CI port found a new, remote CI node, the CI port did not complete the processing of a "virtual-circuit-open" command within the permitted virtual-circuit timeout period. The local CI port is either hung, offline, or saturated with activity, or the CI response queue processing CPU remained at or above IPL-8 for an extended period.

User Action: The port restarts automatically. If this error recurs, it indicates problems with port load balancing, system tuning, or faulty hardware or firmware on the CI port.

DOSETVOL, Use SET VOLUME/STRUCTURE_LEVEL and remount.

Facility: MOUNT, Mount Utility

Explanation: This message accompanies the INCONSTRUCT and INCONSET error messages to provide instructions for making the volume or volume set consistent.

User Action: If the volume or volume set is not currently mounted /NOSHARE, dismount the portion that is currently mounted and mount the volume or volume set using /NOSHARE. Then execute the DCL command SET VOLUME /STRUCTURE_LEVEL=5 on the volume or volume set. Later, you can dismount the volume or volume set and mount it as you wish.

FIBREPOLL, scanning for devices through FIBRE port 'port-device'

Facility: IOGEN

Explanation: This message is displayed while IO AUTOCONFIGURE scans a fibre channel bus to locate and configure devices on that bus.

User Action: None.

FTLIOERR, fatal I/O error while trying to access device

Facility: IOGEN

Explanation: During device configuration, OpenVMS was unable to communicate with one of the devices identified during its scan of the I/O buses.

User Action: Verify that your hardware configuration is set up properly and make any necessary corrections. If no configuration problems are noted, contact a Compaq support representative for assistance.

INCONSET, structure level on 'device-name' is inconsistent with volume set

Facility: MOUNT, Mount Utility

Explanation: The file structure level recorded on the volume is not consistent with the structure level of the volume set of which it is a part. The volume is marked as Files–11 structure level 2 while the rest of the volume set is structure level 5, or vice versa. This condition can occur if a system failure occurs during the execution of a SET VOLUME /STRUCTURE_LEVEL=5 command.

This error is fatal for all shareable mounts (for example, MOUNT/SYSTEM and MOUNT/SHARE). It is a warning for /NOSHARE mounts; however, the volume set is locked against modification. That is, existing files can be accessed and updated, but they cannot be extended or deleted.

User Action: If the volume set is not currently mounted /NOSHARE, dismount the portion that is currently mounted and mount the entire set using /NOSHARE. Then execute the DCL command SET VOLUME /STRUCTURE_LEVEL=5 on the volume set. Later, you can dismount the volume set and mount it as you wish.

INCONSTRUCT, inconsistent file structure level on 'device-name'

Facility: MOUNT, Mount Utility

Explanation: The file structure level recorded on the volume is inconsistent. Parts of the volume are identified as Files–11 structure level 2 and other parts are identified as structure level 5. This condition can occur if a system failure occurs during the execution of a SET VOLUME /STRUCTURE_LEVEL=5 command.

This error is fatal for all shareable mounts (for example, MOUNT/SYSTEM and MOUNT/SHARE). It is a warning for /NOSHARE mounts; however, the volume is locked against modification. That is, existing files can be accessed and updated, but they cannot be extended or deleted.

User Action: Mount the volume using /NOSHARE if it is not currently mounted as such. Then execute the DCL command SET VOLUME /STRUCTURE_LEVEL=5 on it. Later, you can dismount the volume and mount it as you wish.

Insuff. Bus Addr. Pool for Init.: Use AUTOGEN W/FEEDBACK (NPAG_BAP_*)

Facility: Cluster Port Driver

Explanation: The device driver failed to initialize because of a failure to allocate bus addressable pool (BAP). Either BAP was not created or it was insufficient.

User Action: To correctly size and position BAP in physical memory, manually invoke @SYS$UPDATE:AUTOGEN using FEEDBACK mode and reboot. For more information, consult the *OpenVMS System Management Utilities Reference Manual: A–L* or AUTOGEN help.

INVBACKLINK, invalid backlink in extension header 'file-number' on relative volume 'volume-number'

Facility: MOUNT, Mount Utility

Explanation: The backlink on an extension file header does not point to a valid location for a primary header.

User Action: Use the DCL command ANALYZE/DISK_STRUCTURE /REPAIR to repair the volume or volume set.

IVALOCLS, invalid HSZ allocation class, 'allocation-class'

Facility: IOGEN

Explanation: The allocation class read from a storage array exceeds the maximum allowed value of 999.

User Action: From the console of the storage array, set an allocation class for the controller that has a positive value of less than 1000.

MSCIOERR, I/O error while trying to access device

Facility: IOGEN

Explanation: During device configuration, OpenVMS was unable to communicate with one of the devices identified during its scan of the I/O buses.

User Action: Verify that your hardware configuration is set up properly and make any necessary corrections. If no configuration problems are noted, contact a Compaq support representative for assistance.

NOALOCLS, no multibus HSZ allocation class

Facility: IOGEN

Explanation: When you put a storage array into multibus failover mode, you must also assign an allocation class to the array controllers. OpenVMS has not obtained a valid allocation class from this device.

User Action: From the console of the storage array, set an allocation class for the array controller.

NODEVID, no identifier defined for device

Facility: IOGEN

Explanation: The device being configured either has been assigned an invalid device ID or it has an invalid or missing worldwide ID.

User Action: Assign a valid device ID to the failing device, following the procedure described in the documentation for the device or storage array. If you still have a problem, try replacing the drive. If the error persists, contact a Compaq support representative for assistance.

NOLUNS, no logical units found by array controller

Facility: IOGEN

Explanation: The storage array being configured by OpenVMS has reported that it contains no storage devices.

User Action: If you wish, add drives to the storage array.

Port incompatible with nonzero system base PA.

Facility: Cluster Port Driver

Explanation: You attempted to use an older non-PCI device, such as an XMI-based CIXCD or KFMSB, on an Alpha system whose main memory does not begin at physical address zero (PA0).

User Action: Use CIXCD or KFMSB adapters only in systems with system memory based at physical address zero (0). Use adapters such as CIPCA or KFPSA in systems with system memory based above physical address zero.

SCSIPOLL, scanning for devices through SCSI port 'port-device'

Facility: IOGEN

Explanation: This message is displayed while IO AUTOCONFIGURE scans a SCSI bus to locate and configure devices on that bus.

User Action: None.

Index

E

F

G

H

I

J

K

L

M

N

O

P

R

S

T

U

V

W

X

Y