OpenVMS Version 7.1 New Features Manual

Order Number: AA–QSBFB–TE

November 1996

This manual describes the new features of the OpenVMS VAX Version 7.1 and the OpenVMS Alpha Version 7.1 operating systems.

Revision/Update Information:	This is a new manual.
Software Version:	OpenVMS Alpha Version 7.1 OpenVMS VAX Version 7.1

Digital Equipment Corporation
Maynard, Massachusetts

November 1996

Digital Equipment Corporation makes no representations that the use of its products in the manner described in this publication will not infringe on existing or future patent rights, nor do the descriptions contained in this publication imply the granting of licenses to make, use, or sell equipment or software in accordance with the description.

Possession, use, or copying of the software described in this publication is authorized only pursuant to a valid written license from Digital or an authorized sublicensor.

Digital conducts its business in a manner that conserves the environment and protects the safety and health of its employees, customers, and the community.

© Digital Equipment Corporation 1996. All rights reserved.

The following are trademarks of Digital Equipment Corporation: AlphaServer, AlphaStation, ATMWORKS, Bookreader, CI, DEC Ada, DEC Fortran, DEC VTX, DECchip, DECdirect, DECevent, DECmcc, DECmigrate, DECnet, DECserver, DECswitch, DECthreads, DECUS, DECwindows, Digital, Digital UNIX, GIGAswitch HSC, HSJ, HSZ, InfoServer, LAT, MicroVAX, MSCP, OpenVMS, OpenVMS Cluster, PATHWORKS, POLYCENTER, StorageWorks, TA, ThinWire, TMSCP, TURBOchannel, ULTRIX, VAX, VAX C, VAXcluster, VMS, VMScluster, VT, XMI, and the DIGITAL logo.

The following are third-party trademarks:

Jnet is a registered trademark of Joiner Associates, Inc.

MEMORY CHANNEL is a trademark of Encore Computer Corporation.

Mosaic is a trademark of the University of Illinois.

Motif is a registered trademark of Open Software Foundation, Inc.

MultiNet is a registered trademark of TGV, Inc.

NetView is a registered trademark of International Business Machines Corporation.

NT and Windows NT are trademarks, and Visual C++, Win32, Windows, and Windows 95 are registered trademarks of Microsoft Corporation.

OSI is a registered trademark of CA Management, Inc.

POSTSCRIPT is a registered trademark of Adobe Systems Incorporated.

UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company Ltd.

X/Open is a trademark of X/Open Company Limited.

X Window System is a trademark of the Massachusetts Institute of Technology.

All other trademarks and registered trademarks are the property of their respective holders.

ZK6480

The OpenVMS documentation set is available on CD–ROM.

Contents

Preface ix

1 Summary of Version 7.1 New Features

2 General User Features

2.1 DCL Commands 2–1
2.1.1 PIPE Command 2–1
2.1.2 PPPD Command (Alpha Only) 2–1
2.2 OpenVMS Freeware CD–ROM 2–2
2.3 Internet Support: Enhanced OpenVMS Internet Product Suite 2–2
2.3.1 Overview of the Web Servers 2–2
2.3.2 List of Suite Components 2–4
2.4 Point-to-Point Protocol Support (Alpha Only) 2–6

3 System Management Features

3.1 ATM Local Area Networks: Managing with LANCP (Alpha Only) 3–1
3.1.1 LAN Emulation Over an ATM Network 3–1
3.1.2 Classical IP Over an ATM Network 3–2
3.1.3 Managing LAN Emulation Over an ATM Network with LANCP Commands 3–2
3.1.3.1 DEFINE DEVICE Qualifiers for LAN Emulation 3–2
3.1.3.2 LIST DEVICE Qualifiers for LAN Emulation 3–4
3.1.3.3 SET DEVICE Qualifiers for LAN Emulation 3–5
3.1.3.4 SHOW DEVICE Qualifiers for LAN Emulation 3–6
3.1.4 Managing Classical IP Over an ATM Network with LANCP Commands 3–6
3.1.4.1 DEFINE DEVICE Qualifiers for CLIP 3–7
3.1.4.2 LIST DEVICE Qualifiers for CLIP 3–9
3.1.4.3 SET DEVICE Qualifiers for CLIP 3–9
3.1.4.4 SHOW DEVICE Qualifiers for CLIP 3–11
3.2 Backup Utility: /[NO]UNSHELVE Qualifier 3–11
3.3 DECamds: Right to Use Provided with OpenVMS License 3–12
3.4 Dump File Features 3–12
3.4.1 Dump File Off the System Disk (DOSD) 3–12
3.4.2 Error Log Dump File (Alpha Only) 3–12
3.4.3 Selective System Dumps: Fine-Tuning the Order in Which Processes Are Written (Alpha Only) 3–13
3.5 External Authentication Support 3–13
3.5.1 Enabling External Authentication 3–13
3.5.2 External Authentication Over a Network 3–14
3.5.3 Setting a New Password 3–14

3.5.4 How External Authentication Works 3–14
3.5.4.1 User Name Mapping and Password Verification 3–15
3.5.4.2 Password Synchronization 3–15
3.6 Fast Ethernet LANs (Alpha Only) 3–15
3.7 High SCSI IDs on Wide Adapters (Alpha Only) 3–16
3.8 LAN Control Program (LANCP) Utility: New Commands 3–17
3.8.1 CONVERT DEVICE_DATABASE 3–17
3.8.2 CONVERT NODE_DATABASE 3–17
3.9 License Management Facility (LMF) Features 3–17
3.10 Networking Support 3–18
3.10.1 DECnet-Plus for OpenVMS 3–19
3.10.1.1 New Features of DECnet-Plus 3–19
3.10.1.2 Upgrading to DECnet-Plus 3–20
3.10.2 Digital TCP/IP Services for OpenVMS 3–20
3.11 OpenVMS Cluster Features 3–21
3.11.1 MEMORY CHANNEL Interconnect 3–21
3.11.2 New CI-to-PCI Adapter, CIPCA 3–22
3.11.3 New LAN$POPULATE Support 3–22
3.11.4 Lock Manager: Increased Quotas and Limits 3–23
3.11.5 New CLUSTER_CREDITS System Parameter 3–23
3.11.6 New Cluster Configuration Options 3–24
3.11.7 New SCSI Adapter Support 3–24
3.11.8 New SCSI Device Naming Scheme 3–24
3.11.9 SCSI Support for 16 Devices 3–25
3.11.10 OpenVMS Cluster Compatibility Kit 3–25
3.11.11 Warranted and Migration Support for OpenVMS Cluster Configurations 3–25
3.12 OpenVMS Management Station: Offers Enhanced Printer Management 3–26
3.13 OpenVMS Volume Shadowing: Minimerge Capability for System Disks 3–26
3.14 POLYCENTER Software Installation Utility: New Subcommands and Displays 3–27
3.14.1 EXTRACT FILE Subcommand 3–27
3.14.2 EXTRACT PDF Subcommand 3–27
3.14.3 EXTRACT PTF Subcommand 3–28
3.14.4 LIST Subcommand 3–28
3.14.5 SHOW UTILITY Subcommand 3–28
3.15 SYSMAN Utility: New Qualifiers 3–28
3.15.1 DO/CONFIRM Qualifier 3–28
3.15.2 DO/PAUSE Qualifier 3–29
3.15.3 IO REBUILD Command (Alpha Only) 3–29
3.15.4 PARAMETERS SHOW/PAUSE Qualifier 3–29
3.16 System Parameters: New and Changed 3–30
3.16.1 CLUSTER_CREDITS Parameter 3–30
3.16.2 DEVICE_NAMING Parameter (Alpha Only) 3–30
3.16.3 GBLPAGFIL and GBLPAGES Parameters (Alpha Only) 3–30
3.16.4 LAN_FLAGS Parameter (Alpha Only) 3–30
3.16.5 MC_SERVICES Parameters 3–31
3.16.6 MSCP_CMD_TMO Parameter 3–31
3.16.7 Reclamation Parameters 3–32
3.16.7.1 NPAG_AGGRESSIVE Parameter (Alpha Only) 3–32
3.16.7.2 NPAG_GENTLE Parameter (Alpha Only) 3–32
3.16.7.3 NPAG_INTERVAL Parameter (Alpha Only) 3–32

3.16.8 New RMS Parameters 3–32
3.16.8.1 RMS_HEURISTIC Parameter 3–32
3.16.8.2 RMS_DFLRL Parameter 3–33

4 Programming Features

4.1 Very Large Memory Management Features (Alpha Only) 4–1
4.2 Backup Utility: New Application Programming Interface (API) 4–1
4.3 Byte and Word Instruction Emulator: Now Part of the Installation Kit (Alpha Only) 4–2
4.4 C Function Prototypes: Documented in the System Services Routines (Alpha Only) 4–2
4.5 DECwindows X11 Display Server: Upgraded to Version 11, Release 6 . . . 4–3
4.6 Ethernet Media Types: Now Selected from the Console (Alpha Only) 4–3
4.7 Fast Ethernet Adapters: Support for the DE500-AA Adapter (Alpha Only) 4–4
4.7.1 Enabling Auto Negotiation with the Console Environment Variable 4–5
4.7.2 Enabling Auto Negotiation with LANCP 4–5
4.8 Kernel Threads Features 4–6
4.8.1 Linker Utility: New /THREADS_ENABLE Qualifier 4–6
4.8.2 THREADCP Tool (Alpha Only) 4–8
4.9 LAN Emulation: Over an ATM Network (Alpha Only) 4–8
4.9.1 Components 4–9
4.9.2 ATM Adapters 4–10
4.9.3 Topography 4–10
4.9.4 Supporting LAN Emulation Over ATM 4–10
4.9.5 Enabling SONET/SDH 4–11
4.9.6 Booting MOP 4–12
4.9.7 Configuring an Emulated LAN (ELAN) 4–12
4.9.8 Configuring Digital ATM Hardware 4–12
4.9.8.1 Configuring a GIGAswitch/ATM for ELAN 4–13
4.9.9 Configuring Digital TCP/IP Services for OpenVMS Over an Emulated LAN 4–14
4.9.10 Configuring PATHWORKS Over an Emulated LAN 4–14
4.10 MACRO–32 Compiler Built-Ins (Alpha Only) 4–14
4.11 OpenVMS Debugger Features 4–16
4.11.1 Debug Fixup Table (DFT) Processing: Optimized to Improve Debugger Startup Time 4–16
4.11.2 Debugger Commands: New and Changed 4–16
4.11.2.1 Byte and Word Emulation: Enhanced Support (Alpha Only) 4–16
4.11.2.2 DUMP Command 4–17
4.11.2.3 Enhanced CALL Command 4–17
4.11.2.4 SET BREAK/HANDLER Qualifier 4–17
4.11.2.5 SET TERMINAL/WRAP Qualifier 4–18
4.11.3 DECwindows Motif Interface: Now Has Two Wait Cursors 4–18
4.11.4 Debugger Symbol File (.DSF) File Support (Alpha Only) 4–18
4.11.5 Fortran Debugging: Enhanced Operator and Variable Support 4–18
4.11.6 Register Displays: Predefined in Screen Mode 4–18
4.12 OpenVMS SCSI Device Drivers: Enhanced for OpenVMS Version 7.1 4–19
4.13 SCSI Skip Files: New Modifier for SCSI IO$_SKIPFILE Function 4–19
4.14 System Dump Analyzer (SDA) Features (Alpha Only) 4–20

4.14.1 New Commands 4–20
4.14.1.1 SET ERASE_SCREEN Command 4–20
4.14.1.2 SHOW ADDRESS Command 4–20
4.14.1.3 SHOW BUGCHECK Command 4–21
4.14.1.4 SHOW GLOBAL_SECTION_TABLE Command 4–22
4.14.1.5 SHOW GSD Command 4–23
4.14.1.6 SHOW RMD Command 4–24
4.14.1.7 SHOW WORKING_SET_LIST Command 4–25
4.14.1.8 VALIDATE PFN_LIST Command 4–25
4.14.2 New Qualifiers, Options, and Parameters 4–26
4.14.2.1 EVALUATE Qualifier 4–26
4.14.2.2 FORMAT Qualifier 4–26
4.14.2.3 READ Qualifiers 4–26
4.14.2.4 SEARCH Qualifier 4–27
4.14.2.5 SET FETCH Qualifiers 4–27
4.14.2.6 SHOW LAN Qualifiers 4–27
4.14.2.7 SHOW PFN_DATA Qualifiers 4–30
4.14.2.8 SHOW POOL Qualifier 4–31
4.14.2.9 SHOW PROCESS Qualifiers 4–31
4.14.2.10 SHOW PROCESS Options 4–32
4.14.3 Arithmetic and Logical Operators 4–32
4.14.4 Syntax for Displaying Range of Page Table Entries 4–32
4.14.5 Behavior for the READ Command 4–33
4.14.6 New DUMPSTYLE File Features 4–33
4.15 System Services Features 4–34
4.15.1 Changed Services: Enhanced to Support 64-Bit Addressing (Alpha Only) 4–34
4.15.2 New System Services 4–34
4.15.2.1 $AVOID_PREEMPT 4–34
4.15.2.2 $SETUP_AVOID_PREEMPT 4–35
4.15.2.3 $FREE_USER_CAPABILITY 4–35
4.15.2.4 $GET_USER_CAPABILITY 4–35

5 Documentation Features

5.1 Available Documentation for OpenVMS Version 7.1 5–1
5.2 Select Documentation Offered in HTML Format 5–1
5.3 DECnet-Plus for Version 7.1 Documentation 5–1
5.4 Device Support Documentation 5–2
5.5 Migration Documentation 5–3
5.6 Archived Documentation 5–3

6 Automatically Configuring Devices for OpenVMS Alpha

6.1 Overview of Device Configuration 6–1
6.2 Using File-Based Autoconfiguration 6–2
6.2.1 Adding Descriptions to USER_CONFIG.DAT 6–2
6.2.2 Configuration File Syntax 6–3
6.2.3 Device Descriptions 6–3
6.2.4 Rebuilding the USER_CONFIG.DAT File 6–6
6.3 Supported Buses for User Devices 6–7
6.3.1 ISA Configuration 6–7
6.3.1.1 Configuring ISA Devices on the ISA Bus 6–7
6.3.1.2 Configuring ISA devices on the EISA Bus 6–7

6.4 SYS$MANAGER:ISA_CONFIG.DAT Unsupported in Future Releases . . . 6–8
6.5 Error Information 6–11

A New OpenVMS System Messages

Index

Figures

3–1 OpenVMS Cluster Version Pairings 3–26
4–1 Emulated LAN Topography 4–11

Tables

1–1 Summary of OpenVMS VAX and OpenVMS Alpha Version 7.1 Software Features 1–1
2–1 Internet Product Suite Components 2–4
3–1 LAN ATM Devices 3–1
3–2 LAN Media Characteristics 3–1
3–3 DEFINE DEVICE Command Qualifiers for LAN Emulation 3–2
3–4 LIST DEVICE Command Qualifiers for LAN Emulation 3–4
3–5 SET DEVICE Command Qualifiers for LAN Emulation 3–5
3–6 SHOW DEVICE Command Qualifiers for LAN Emulation 3–6
3–7 DEFINE DEVICE Command Qualifiers for CLIP 3–7
3–8 LIST DEVICE Command Qualifiers for CLIP 3–9
3–9 SET DEVICE Command Qualifiers for CLIP 3–9
3–10 SHOW DEVICE Command Qualifiers for CLIP 3–11
3–11 Fast Ethernet Cabling 3–16
3–12 LICENSE Enhancements 3–18
4–1 Components of LAN Emulation Over an ATM Network 4–9
4–2 ATM Adapters 4–10
4–3 New Alpha Instruction Built-Ins 4–14
4–4 SET ERASE_SCREEN Parameters 4–20
4–5 SHOW ADDRESS Parameter 4–20
4–6 SHOW ADDRESS Qualifier 4–20
4–7 SHOW BUGCHECK Parameters 4–21
4–8 SHOW BUGCHECK Qualifier 4–22
4–9 SHOW GLOBAL_SECTION_TABLE Qualifier 4–22
4–10 SHOW GSD Qualifiers 4–23
4–11 SHOW RMD Qualifiers 4–24
4–12 Options for the SHOW WORKING_SET_LIST Command 4–25
4–13 VALIDATE PFN_LIST Qualifiers 4–25
4–14 EVALUATE Command New Qualifier 4–26
4–15 FORMAT Command New Qualifier 4–26
4–16 READ Command New Qualifiers 4–26
4–17 SEARCH Command New Qualifier 4–27
4–18 SET FETCH Command New Qualifiers 4–27

4–19	SHOW LAN Command New Qualifiers	4–27
4–20	SHOW PFN_DATA Command New Qualifiers	4–30
4–21	Command Options with the /COLOR Qualifier	4–31
4–22	SHOW POOL Command New Qualifier	4–31
4–23	SHOW PROCESS /PAGE_TABLES Command New Qualifiers	4–32
4–24	SHOW PROCESS /WORKING_SET_LIST Command New Options	4–32
4–25	New Unary Operators	4–32
4–26	New Range Page Table Syntax	4–33
5–1	DECnet Phase IV Manuals	5–2
5–2	OpenVMS Version 7.1 Device Support Books	5–3
5–3	Manuals Archived for Version 7.1	5–4
6–1	Keyword Summary	6–6
6–2	ISA_CONFIG.DAT Keywords and Equivalents	6–8

Preface

Intended Audience

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

This document contains descriptions of the new features for Version 7.1 of the OpenVMS VAX and OpenVMS Alpha operating systems. For information about how some of the new features might affect your system, read the *OpenVMS Version 7.1 Release Notes* before you install, upgrade, or use Version 7.1.

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 describes new features of the OpenVMS documentation set; this includes both online and printed material.
- Chapter 6 describes a new method of automatically configuring third-party hardware devices.
- Appendix A describes new or changed messages from the Help Message database.

Related Documents

Refer to the following documents for detailed information about the software features described in this manual:

- *DECnet-Plus Application Installation and Advanced Configuration*
- *DECnet-Plus for OpenVMS Release Notes*
- *DECnet-Plus Installation and Basic Configuration*
- *DECnet-Plus Introduction and User's Guide*
- *OpenVMS Alpha System Dump Analyzer Utility Manual*
- *OpenVMS Alpha Version 7.1 Upgrade and Installation Manual*
- *OpenVMS DCL Dictionary*
- *OpenVMS Debugger Manual*

- *OpenVMS Guide to System Security*
- *OpenVMS I/O User's Reference Manual*
- *OpenVMS License Management Utility Manual*
- *OpenVMS System Management Utilities Reference Manual*
- *OpenVMS System Manager's Manual*
- *OpenVMS System Manager's Manual: Tuning, Monitoring, and Complex Systems*
- *OpenVMS System Messages: Companion Guide for Help Message Users*
- *OpenVMS System Services Reference Manual*
- *OpenVMS Utility Routines Manual*
- *OpenVMS VAX Version 7.1 Upgrade and Installation Manual*
- *OpenVMS Version 7.1 Release Notes*

For additional information on the Open Systems Software Group (OSSG) products and services, access the Digital OpenVMS World Wide Web site. Use the following URL:

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

Reader's Comments

Digital 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	603 881-0120, Attention: OSSG Documentation, ZKO3-4/U08
Mail	OSSG Documentation Group, ZKO3-4/U08 110 Spit Brook Rd. Nashua, NH 03062-2698

How To Order Additional Documentation

Use the following table to order additional documentation or information. If you need help deciding which documentation best meets your needs, call 800-DIGITAL (800-344-4825).

Telephone and Direct Mail Orders

Location	Call	Fax	Write
U.S.A.	DECdirect 800–DIGITAL 800–344–4825	Fax: 800–234–2298	Digital Equipment Corporation P.O. Box CS2008 Nashua, NH 03061
Puerto Rico	809–781–0505	Fax: 809–749–8300	Digital Equipment Caribbean, Inc. 3 Digital Plaza, 1st Street, Suite 200 P.O. Box 11038 Metro Office Park San Juan, Puerto Rico 00910–2138
Canada	800–267–6215	Fax: 613–592–1946	Digital Equipment of Canada, Ltd. Box 13000 100 Herzberg Road Kanata, Ontario, Canada K2K 2A6 Attn: DECdirect Sales
International	—	—	Local Digital subsidiary or approved distributor
Internal Orders	DTN: 264–4446 603–884–4446	Fax: 603–884–3960	U.S. Software Supply Business Digital Equipment Corporation 10 Cotton Road Nashua, NH 03063–1260

ZK–7654A–GE

Conventions

The name of the OpenVMS AXP operating system has been changed to the OpenVMS Alpha operating system. Any references to OpenVMS AXP or AXP are synonymous with OpenVMS Alpha or Alpha.

VMScluster systems are now referred to as OpenVMS Cluster systems. Unless otherwise specified, references to OpenVMS Clusters or clusters in this manual are synonymous with VMSclusters.

In this manual, every use of DECwindows and DECwindows Motif refers to DECwindows Motif for OpenVMS software.

The following conventions are also 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* or GOLD *x*	A sequence such as PF1 *x* or GOLD *x* indicates that you must first press and release the key labeled PF1 or GOLD and then press and release another key or a pointing device button. GOLD key sequences can also have a slash (/), dash (–), or underscore (_) as a delimiter in EVE commands.
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.)
. . .	Horizontal ellipsis points in examples indicate 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.
. . .	Vertical ellipsis points indicate 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, if you choose more than one option, you must enclose the choices in parentheses.
[ ]	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, braces indicate a required choice of options; you must choose one of the options listed.
text style	This text style represents the introduction of a new term or the name of an argument, an attribute, or a reason. This style is also used to show user input in Bookreader versions of the manual.

italic text	Italic text indicates important information, complete titles of manuals, or variables. Variables include information that varies in system messages (Internal error *number*), in command lines (/PRODUCER=*name*), and in command parameters in text (where *device-name* contains up to five alphanumeric characters).
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 type`	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.
-	A hyphen in code examples indicates that additional arguments to the request are provided on the line that follows.
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 Version 7.1 New Features

The main features of OpenVMS Version 7.1 provide improved OpenVMS Cluster performance, options for improving the performance of very large database applications, easier system management, and the ability to easily connect OpenVMS systems to the Internet.

Table 1–1 summarizes each feature provided by OpenVMS Alpha and OpenVMS VAX Version 7.1 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 7.1 Software Features

General User Features	
DCL commands	Two new DCL commands have been added: • PIPE—Enables you to perform UNIX command parsing from the DCL command line. • PPPD (Alpha systems only)—Invokes the new Point-to-Point Protocol utility (PPPD).
OpenVMS freeware	The OpenVMS Freeware CD–ROM contains free software tools and utilities that aid OpenVMS software development and system management.
Internet support	The OpenVMS Internet Product Suite offers a convenient and comprehensive portfolio of Digital and third-party internet products, tools, and utilities you can use to set up and maintain intranet or Internet sites.
Point-to-Point Protocol support (Alpha systems only)	OpenVMS Alpha Version 7.1 offers a new utility and device driver that supports the Point-to-Point Protocol (PPP), which is the widely accepted Internet protocol for serial line communications.

(continued on next page)

Table 1–1 (Cont.) Summary of OpenVMS VAX and OpenVMS Alpha Version 7.1 Software Features

System Management Features	
Asynchronous transfer mode (ATM) LAN management (Alpha systems only)	OpenVMS Alpha Version 7.1 now supports managing LANs and Classical IP (CLIP) over ATM networks using LANCP commands.
New Backup utility qualifier	The /[NO]UNSHELVE qualifier controls whether the Backup utility designates files from a BACKUP save operation as unshelved or shelved.
DECamds utility	The right to use DECamds is now part of the OpenVMS base operating system license. DECamds provides system resource monitoring, investigation aid, and correction capability for multiple systems from a centralized DECwindows display.
Dump file features	OpenVMS Version 7.1 includes the following new dump file features: • Dump file off the system disk (DOSD) The system dump file can now be written to devices other than the system disk. • Error log dump file (Alpha systems only) A new file, in addition to the system dump file, that Alpha systems use during a system crash to record the contents of error log buffers. • User-definable ‘‘key’’ processes in selective system dumps (Alpha systems only) A system manager can now designate specific processes (key processes) as having priority over other processes in a selective dump.
External authentication support	Allows users to log in (or sign on) to an OpenVMS system at the OpenVMS login prompts with user IDs and passwords from an external authenticator (such as LAN Manager).
Fast Ethernet LANs (Alpha systems only)	OpenVMS Alpha Version 7.1 supports Fast Ethernet LANs that raise the data transmission rates from 10Mb/s to 100Mb/s.
High SCSI IDs (Alpha systems only)	OpenVMS Alpha Version 7.1 supports up to 16 SCSI devices per wide SCSI bus. This enables devices that support the wide SCSI bus option to use existing hardware more fully for device identification.
LAN Control Program (LANCP) utility feature	The following new LANCP commands have been added: CONVERT DEVICE_DATABASE CONVERT NODE_DATABASE
License Management Facility (LMF) features	LMF includes the following new features for OpenVMS Version 7.1: • Ability to load multiple license types for one product on the same node • Use of reservation lists to reserve allocation units for specific users • Ability to control the order of license loading

(continued on next page)

Table 1–1 (Cont.) Summary of OpenVMS VAX and OpenVMS Alpha Version 7.1 Software Features

System Management Features	
Networking support	Whether you require TCP/IP, OSI, or DECnet, OpenVMS now enables you to simply choose the protocol or combination of protocols that work best for your network.Also, DECnet-Plus and Digital TCP/IP Services for OpenVMS are now part of the OpenVMS Alpha installation and upgrade procedures.
OpenVMS Cluster features	OpenVMS Version 7.1 includes the following new cluster features: • MEMORY CHANNEL interconnect (Alpha systems only) MEMORY CHANNEL is a new, high-performance (100 Mb/s) cluster interconnect technology for PCI-based Alpha systems. MEMORY CHANNEL offers significant performance increases over the traditional interconnect technology, such as the CI. • CI-to-PCI (CIPCA) adapter (Alpha systems only) With the release of OpenVMS Alpha Version 6.2–1H2, Digital introduced a CI-to-PCI adapter (CIPCA) that was developed in partnership with CMD Technologies. With this adapter, Alpha servers that contain a combination of the PCI and the EISA buses can now connect to the CI. • LAN$POPULATE additional functions LAN$POPULATE now provides support for migrating satellite booting from DECnet-Plus to LANCP and support for cleaning out the DECnet database when migrating from DECnet Phase IV. • Increased lock manager quotas and limits Increases in the lock manager quotas and the elimination of some limits are included in this release. • New options (Make Root and Delete Root) in CLUSTER_ CONFIG.COM and CLUSTER_CONFIG_LAN.COM • New adapter support for SCSI OpenVMS Cluster configurations OpenVMS Version 7.1 supports the KZPSA and KZTSA adapters. These SCSI adapters connect a PCI bus (KZPSA) or a TURBOchannel bus (KZTSA) to a single fast-wide differential SCSI bus. • New SCSI device naming scheme Port allocation classes are a new method for naming SCSI devices attached to Alpha systems in an OpenVMS Cluster. The new device names can be served to OpenVMS VAX systems via the MSCP server. Port allocation classes are designed to alleviate naming conflicts. • Support for coexistence with OpenVMS Versions 6.2 and 7.0 in mixed-architecture cluster configurations

(continued on next page)

Table 1–1 (Cont.) Summary of OpenVMS VAX and OpenVMS Alpha Version 7.1 Software Features

System Management Features	
OpenVMS Management Station	You can now manage a wide range of printers and print queues across multiple OpenVMS Cluster systems and OpenVMS nodes more easily. From a new, easy-to-use Windows interface, you can create, delete, and manage a printer and its related queues, as well as perform print job management.
OpenVMS Volume Shadowing feature	The OpenVMS Volume Shadowing mini-merge capability has been enabled on CI- and DSSI-based system disks. This provides faster merge operations for shadowed system disks by eliminating the I/O overhead associated with a full merge operation.
POLYCENTER Software Installation utility features	Several PRODUCT subcommands have been enhanced, and the following new PRODUCT subcommands have been added: EXTRACT FILE EXTRACT PDF EXTRACT PTF LIST SHOW UTILITY
SYSMAN utility features	A new command, IO REBUILD, has been added as well as the following new qualifiers: DO/CONFIRM DO/PAUSE PARAMETERS SHOW/PAUSE
System parameters	OpenVMS Version 7.1 includes the following new system parameters: CLUSTER_CREDITS DEVICE_NAMING (Alpha systems only) GBLPAGFIL GBLPAGES LAN_FLAGS (Alpha systems only) MC_SERVICES_P0...9 MSCP_CMD_TMO NPAG_AGGRESSIVE (Alpha systems only) NPAG_GENTLE (Alpha systems only) NPAG_INTERVAL (Alpha systems only)
Programming Features	
Very Large Memory management features (Alpha systems only)	OpenVMS Alpha Version 7.1 now includes extended Very Large Memory (VLM) management capabilities such as memory-resident global sections, shared page tables, and a reserved memory registry.
New Backup utility interface	The Backup utility now includes a new callable application programming interface (API).
Byte and Word Instruction Emulator (Alpha systems only)	Incorporated into the OpenVMS Alpha Version 7.1 installation kit. This emulator allows the execution of byte and word instructions on Alpha systems that contain DECchip 21064 or 21164.
C function prototypes (Alpha systems only)	Now documented in the *OpenVMS System Services Reference Manual*.

(continued on next page)

Table 1–1 (Cont.) Summary of OpenVMS VAX and OpenVMS Alpha Version 7.1 Software Features

Programming Features	
DECwindows X11 Display Server	Upgraded to X11 Release 6 and includes an updated X Logical Font Description along with the following new server extensions: XTest XC-MISC BIG-REQUESTS
Ethernet media types (Alpha systems only)	The SYS$EWDRIVER LAN device driver now uses a console environment variable to select the proper media connection.
Fast Ethernet adapter support (Alpha systems only)	The DE500-AA PCI Ethernet adapter provides 100 Mb/s Ethernet support.
Kernel threads features	A new linker qualifier, /THREADS_ENABLE, has been added that allows you to control kernel threads features on a per-image basis. Also, OpenVMS Alpha includes the THREADACP tool which enables threads features for an image that will not be rebuilt.
LAN emulation over an ATM network (Alpha systems only)	OpenVMS Alpha Version 7.1 supports Ethernet and Token Ring emulation over an ATM network.
MACRO–32 compiler support for byte and word instructions (Alpha systems only)	The MACRO–32 compiler provides new built-ins that allow the inclusion of byte and word instructions.
OpenVMS Debugger features	The following new debugger features have been added: • Enhanced support for byte and word emulation allows you to set breakpoints to suspend debugger execution whenever the operating system emulates a byte or word instruction. • New predefined register displays in screen mode allow you to display all Alpha registers, Alpha floating-point registers, or Alpha general-purpose registers. • New output message control in screen mode allows you to specify the point at which messages will wrap to the next line. • The DUMP command allows you to display the contents of memory in a manner similar to the DCL command DUMP. • .DSF File Support (Alpha systems only) allows you to debug programs that have symbol information in associated .DSF files.

(continued on next page)

Table 1–1 (Cont.) Summary of OpenVMS VAX and OpenVMS Alpha Version 7.1 Software Features

Programming Features	
SCSI device driver enhancements	The SCSI device drivers now work with a wider range of non-Digital SCSI devices.
New SCSI skip file function modifier	The IO$M_ALLOWFAST modifier can be used with the IO$_SKIPFILE function to provide better performance on SCSI tape drives that support the SCSI space-by-file-marks command and the SCSI read position command.
New Alpha System Dump Analyzer (SDA) features (Alpha systems only)	OpenVMS Alpha SDA Version 7.1 supports numerous new commands, parameters, qualifiers, and new dump style features.
System services features	Several system services have been enhanced to support 64-bit addressing on OpenVMS Alpha systems, and the following new system services have been added: $AVOID_PREEMPT $SETUP_AVOID_PREEMPT $FREE_USER_CAPABILITY $GET_USER_CAPABILITY
Documentation Features	
OpenVMS documentation	The OpenVMS Version 7.1 documentation offering provides the following features: • All documents available in print for Version 7.1 • Select OpenVMS documentation available in HTML format • Special DECnet-Plus Starter Kit and documentation that is shipped with the OpenVMS documentation • Enhanced device support documentation • Consolidated OpenVMS VAX to OpenVMS Alpha migration documentation
Automatically Configuring Hardware Devices for OpenVMS Alpha	
File-based autoconfiguration (Alpha systems only)	A new method for configuring third-party hardware devices using ASCII text files.

(continued on next page)

Table 1–1 (Cont.) Summary of OpenVMS VAX and OpenVMS Alpha Version 7.1 Software Features

OpenVMS System Messages	
New system messages	New messages for the following utilities: BACKUP, Backup Utility CLI, Command Language Interpreter (DCL) Cluster Port Driver LICENSE, License Management Utility LOGIN, Login Procedure MC, MEMORY CHANNEL MOUNT, Mount Utility PPPD, Point-to-Point Protocol Utility SDA, System Dump Analyzer SDP, Set Dump Priority Utility SET, SET PASSWORD Command SHADOW, Volume Shadowing Software SHADOW_SERVER, Volume Shadowing Server SMI, System Management Integrator/Server SYSINIT, System Initialization SYSMAN, System Management Utility SYSTEM, System Services UAF, Authorize Utility

2
General User Features

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

2.1 DCL Commands

This section describes the new DCL commands for OpenVMS Version 7.1.

2.1.1 PIPE Command

OpenVMS Version 7.1 introduces some popular UNIX command shell processing functions. With the PIPE command, you can quickly create complex command processing statements that combine the following operations in a single DCL command:

- Pipelining
- Input/output redirection
- Multiple and conditional command execution
- Background processing

For example, the following PIPE command runs a test program, sorts the results, and compares those results to a benchmark file. This single command statement simplifies processing and eliminates intermediate files.

```
$ PIPE RUN TEST.EXE | SORT/SPECIFICATION=TEST.SRT SYS$INPUT SYS$OUTPUT -
| DIFF SYS$INPUT  TEST.BENCHMARK
```

This style of command processing also supports the development and use of Internet software, which often expects some form of pipeline command parsing to be present on both host and target systems.

For a more detailed description of this command, see the *OpenVMS DCL Dictionary: N–Z*.

2.1.2 PPPD Command (Alpha Only)

The DCL command PPPD invokes the Point-to-Point Protocol utility, new with OpenVMS Alpha Version 7.1. This utility enables you to set up both permanent and temporary serial network connections between an OpenVMS Alpha workstation and a specific local or remote host. However, the host must support the Point-to-Point Protocol (PPP) internet communication standard (STD 51) established by the Internet Architecture Board (IAB).

For more detailed information about the PPPD command, see the *OpenVMS DCL Dictionary: N–Z*. For a description of the Point-to-Point Protocol utility, see Section 2.4.

2.2 OpenVMS Freeware CD–ROM

OpenVMS Engineering is offering the OpenVMS Freeware Version 3.0 CD–ROM with the OpenVMS Version 7.1 distribution kit. It contains free software that runs on OpenVMS Version 6.0 or later and provides you with easy access to public domain software and free internal Digital tools that facilitate software development and system management.

The OpenVMS Freeware CD–ROM features the following:

- All binaries, source code (except for Bliss compilers), and documentation
- An easy-to-use, 4GL-based menu system for viewing the contents of the CD–ROM
- A single, comprehensive online documentation file that provides quick browsing capability
- A Files–11 format that makes the CD–ROM readable on OpenVMS systems

Disclaimer

The OpenVMS Freeware CD–ROM is provided ‘‘as is’’ without a warranty. Digital imposes no restrictions on its distribution nor on the redistribution of its contents. Be aware, however, that some of the packages on the disc can carry restrictions on their use imposed by the original authors. Therefore, you should carefully read the documentation that accompanies each product.

Specifically, the guidelines for the software on the OpenVMS Freeware CD–ROM are as follows:

- No license fees are necessary for its use.
- The software must run on OpenVMS Version 6.0 or later.
- Digital does not provide support services for this software, fix the software, or guarantee that it works correctly.
- As a user, you are free to adapt the source files to your needs.

2.3 Internet Support: Enhanced OpenVMS Internet Product Suite

The original OpenVMS Internet Product Suite, which debuted in September 1996, provided the opportunity to turn existing OpenVMS VAX or Alpha systems into high performance, highly-available web servers and browsers. Version 1.1 of this product, which is available with OpenVMS Version 7.1, adds the following new components: the Netscape Communications and Commerce servers and an enhanced Purveyor Encrypt web server. This comprehensive and conveniently packaged portfolio of third-party and Digital products gives you an expanded choice of Internet tools to meet your business needs.

2.3.1 Overview of the Web Servers

Digital’s portfolio of OpenVMS Internet products offers you a choice of the commercial HyperText Transport Protocol (HTTP) web servers described in the following sections.

2.3 Internet Support: Enhanced OpenVMS Internet Product Suite

Netscape Communications Server

The Netscape Communications Server Version 1.12 enables you to publish hypermedia documents using the HyperText Markup Language (HTML) on the Internet and internal TCP/IP-based networks (intranet) using HTTP. The server's compatibility with network standards and document formats makes it easy to integrate with other applications and systems in your enterprise.

The easy-to-use graphical user interface (GUI) makes installation, configuration, and management simple and fast. And, it supports industry-standard multimedia document formats for the presentation of rich text, graphics, audio, and video.

Easy-to-navigate hyperlinks connect related information across the network, creating a seamless web that lets you share product information, corporate news, and customer support and service information.

Netscape Commerce Server

The Netscape Commerce Server Version 1.12 provides secure electronic commerce and communications on the Internet and intranets. The Netscape Commerce Server lets you publish hypermedia documents using HTML and deliver them over the Internet and other TCP/IP networks using HTTP. To ensure data security, Netscape Commerce Server provides advanced security features such as server authentication, data encryption, data integrity, and user authorization. Communications are based on open standards including HTML, HTTP, the Common Gateway Interface (CGI), and the Secure Sockets Layer (SSL) protocol.

Purveyor Encrypt WebServer for OpenVMS

The Purveyor Encrypt WebServer for OpenVMS Version 1.2 replaces the original Purveyor WebServer packaged in the first version of the OpenVMS Internet Product Suite. Purveyor Encrypt WebServer combines the reliability, security and scalability of the OpenVMS platform with the graphical ease-of-use and functionality of World Wide Web technologies. Purveyor Encrypt lets you easily web-enable your existing OpenVMS applications, develop a corporate intranet, or create a robust external Web presence all within a secure web environment.

Using Secure Sockets Layer (SSL) encryption and authentication, Purveyor Encrypt ensures the security of access, traffic and requests made to web-enabled OpenVMS systems, which typically house mission-critical data. Purveyor WebServer will also run without encryption services, eliminating unnecessary network overhead.

Purveyor Encrypt runs over TCP/IP for OpenVMS or any other TCP/IP for OpenVMS implementation and can be configured to run in a clustered environment for high availability. Other features include:

- A GUI log viewer that can easily analyze traffic on a web server
- A GUI database wizard that can automatically build HTML FORMS and interact with any ODBC-compliant database
- Virtual servers, which allow more than one web server to be configured
- Integration with the Windows NT user/group database
- A GUI file and link administration application for displaying the structure of web pages and checking the integrity of local and external links
- An integrated proxy server that does not require additional hardware or software to access

2.3.2 List of Suite Components

Table 2–1 lists and describes the products, tools, and utilities available in the OpenVMS Internet Product Suite. For more detailed information, see the online documentation that accompanies the individual products in the OpenVMS Internet Product Suite.

Table 2–1 Internet Product Suite Components

Component	Description
Commercial clients	The Navigator is a World Wide Web browser from Netscape Communication Corporation that supports JavaScript.
	Spyglass Enhanced Mosaic is a World Wide Web browser.
Commercial servers	Netscape Communications Server enables you to publish hypermedia documents using the HyperText Markup Language (HTML) on the Internet and intranets using HTTP.
	Netscape Commerce Server permits secure electronic commerce and communications on the Internet and intranets using Secure Sockets Layer (SSL).
	Purveyor Encrypt is a web server from Process Software Corporation that uses SSL encryption to support secure transactions over the Internet and intranets. It includes a built-in proxy server.
Digital value-added software	The Digital VTX Web Interface is a web interface to Digital's VTX that lets you access your VTX InfoBases using your favorite browser.
	Digital also provides ''Getting started'' web pages and samples to get you started creating HTML pages and browsing the World Wide Web.
Freeware clients	Lynx is a character-cell World Wide Web browser.
	Gopher client is a client that requests and then receives the documents served by a Gopher server on the Internet.
Freeware servers	The (OSU) DECthreads HTTP Server for OpenVMS is a freeware web server from Ohio State University.
	Gopher server is a freeware server that makes documents available on the Internet.
	The CERN Proxy Server is a freeware server that features the TCP/IP-based httpd (HyperText Transfer Protocol Daemon) protocol, caching, and easy Web access from behind firewalls.

(continued on next page)

Table 2–1 (Cont.) Internet Product Suite Components

Component	Description
Freeware UNIX tools	GNU bison is a parser generator (alternative to yacc) by the Andrew Consortium at CMU.
	fgrep, from the Free Software Foundation, is a utility to search for fixed strings.
	flex (Fast Lexical Analyzer Generator) is a tool from the Free Software Foundation for generating programs that perform pattern-matching on text.
	GNU awk for OpenVMS is a search tool from the Free Software Foundation.
	grep is a utility to search for regular expressions.
	perl (Practical Extracting and Report Language) is a scripting language for scanning text files and for implementing web pages.
	GNU sed (Stream Editor) is a UNIX stream editor for OpenVMS from the Free Software Foundation.
	tcl (Tool Command Language) is a stream-based command language.
	tk (Tool Kit) is an X Window System toolkit for tcl.
	yacc (yet another C compiler) is the Berkeley yacc written by the University of California at Berkeley.
	ZIP is a utility that compresses and packages files for OpenVMS VAX and Alpha systems. UNZIP is a utility that lists, tests, or extracts from a ZIP archive on OpenVMS VAX and Alpha systems.
Freeware news readers	ANU-News is a computer-based bulletin board.
	MXRN is an X-based interface to the USENET news system that uses the NNTP remote news server for accessing newsgroups and articles.
	NEWSRDR is an NNTP client program for OpenVMS systems running CMU-Tek TCP/IP, TGV MultiNet, VMS ULTRIX Connection, or any TCP/IP package supporting a Berkeley socket interface.
	VNEWS is a native OpenVMS news reading client that uses the network news transport protocol to access news stored on a remote server.
Freeware viewers	MPEG Video Player is a full-motion video viewer.
	xv is an X Window System graphics viewer.
Freeware mail clients and servers	IUPOP3 is a post office protocol mail server from Indiana University.
	mx (Message Exchange) is electronic mail software for OpenVMS VAX systems running VMS Version 5.0 or later, including OpenVMS Version 6.0, and OpenVMS Alpha systems running OpenVMS Alpha Version 1.0 or later. It supports Internet mail over CMU-OpenVMS TCP/IP, VMS ULTRIX Connection, TGV MultiNet, Process Software's TCPware, and Wollongong's WIN/TCP and PathWay; BITNET mail over Wingra's Jnet; and UUCP mail over DECUS UUCP.
	Pine E-mail is a mail client and newsreader.

(continued on next page)

Table 2–1 (Cont.) Internet Product Suite Components

Component	Description
Freeware HTML tools	HLP2HTML Converter is a tool that converts HLP files to HTML.
	HTML2PS is a tool that converts HTML to PostScript.
	SDML2HTML is a tool that converts SDML to HTML.
	WEBLINT is an HTML parser.
	DEHTML is a utility that removes all HTML markup from a file.
	ENTIFY is a utility that replaces ISO high Latin-1 alphabetic characters with ampersand entities for safe 7-bit transport.
	HTMLCHEK is an HTML syntax checker that uses awk.
	HTMLCHKP is an HTML syntax checker that uses perl.
	HTMLSRPL is a tool that does a string ‘‘search and replace’’ in an HTML file.
	MAKEMENU is a utility that makes a simple menu for HTML files.
	METACHAR is a utility that protects HTML and SGML metacharacters in text that will be included in an HTML file.
	TXT2HTML is a tool that converts plain text to HTML.
Web interfaces	Web interfaces to the DCL help libraries and system monitoring on OpenVMS systems are provided. The interface to OpenVMS system management provides access to such commands as SHOW CLUSTER, SHOW DEVICE, SHOW NETWORK, SHOW QUEUE, and SHOW SYSTEM. The interface also lets you obtain system auditing, accounting, intrusion information, and system logs.

2.4 Point-to-Point Protocol Support (Alpha Only)

The Point-to-Point Protocol (PPP) is an internet layer protocol which is now the widely accepted replacement for the Serial Line Interconnect Protocol (SLIP). PPP support enhances the connectivity and Internet capabilities of OpenVMS Alpha by providing a way to establish a dynamic Internet Protocol (IP) network connection over a serial line — without extensive router or server hardware.

Based on code originated at Carnegie Mellon University, this serial protocol features:

- A way to encapsulate datagrams on a serial link. Specifically, PPP supports an asynchronous link with 8 bits of data and no parity.
- A link control protocol (LCP) that establishes and verifies the data link connection. This protocol enables each host, network, or transport to configure various communication options.

To support this protocol, OpenVMS Alpha Version 7.1 offers a Point-to-Point Protocol utility (PPPD) and related device driver described in the following sections.

Point-to-Point Protocol Utility

The Point-to-Point Protocol utility (PPPD) is a DCL-level tool that you can use to initiate and manage an IP network connection over an asynchronous, serial data line. PPPD extends the networking capability of OpenVMS Alpha by enabling you to:

- Establish temporary, high-speed network connections between remote hosts. This includes both *dial-in* capability from a remote host to an OpenVMS Alpha host and *dial-out* capability from an OpenVMS Alpha host to a remote system or server that supports PPP.
- Establish permanent, low-speed network connections between peers, such as between a laptop computer and an Alpha workstation connected by a serial data line.
- Set and display communication characteristics, such as address compression, flow control, and line speed.

Refer to the *TCP/IP Networking on OpenVMS Systems* for a detailed description of this utility and its related subcommands. Sample PPPD connection procedures are also provided online in an ASCII (.TXT) and PostScript (.PS) file. Once you install the OpenVMS Alpha Version 7.1 kit, you can retrieve either file from the following location:

```
SYS$SYSROOT:[SYSHLP.EXAMPLES.PPPD.DOC]PPP_CONNECT.TXT (or .PS)
```

Note

The Point-to-Point Protocol utility (PPPD) included with OpenVMS Alpha Version 7.1 provides point-to-point networking capability to the base operating system. However, for this utility to be fully functional, your current TCP/IP vendor must also develop and integrate the appropriate callback images into their Internet Protocol (IP) stack.

The Digital TCP/IP Services for OpenVMS team is currently integrating these images into the next version of their product. Digital is also working in tandem with our third-party TCP/IP vendors to provide this functionality; contact your individual TCP/IP vendor for more details.

Point-to-Point Protocol Driver

OpenVMS Alpha Version 7.1 includes a new device driver which provides support for PPP connection management, a state engine, and the link negotiation code. This driver interfaces with the upper layers of the network through the VMS Communications Interface (VCI).

Information about the PPP driver is provided online in both ASCII (.TXT) and PostScript formats (.PS). Once you install the OpenVMS Alpha Version 7.1 kit, you can retrieve either file from the following location:

```
SYS$SYSROOT:[SYSHLP.EXAMPLES.PPPD.DOC]PPP_INTERFACES.TXT (or .PS)
```

3
System Management Features

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

3.1 ATM Local Area Networks: Managing with LANCP (Alpha Only)

OpenVMS Alpha Version 7.1 supports and manages with the LANCP utility the following networking features:

- LAN emulation over an asynchronous transfer mode (ATM) network
- Classical IP (CLIP) over an ATM network

Table 3–1 shows the ATM devices supported by OpenVMS Version 7.1.

Table 3–1 LAN ATM Devices

LAN	Device Examples	Description
ATM	ATMWORKS 350, ATMWORKS 750	The ATMWORKS 350 supports PCI-based systems. The ATMWORKS 750 supports TURBOchannel systems.

3.1.1 LAN Emulation Over an ATM Network

OpenVMS Version 7.1 supports the standards defined by the ATM Forum's LANE Version 1.0 specifications for LAN emulation over an ATM network. By implementing an emulated LAN over an ATM network, you enable a group of ATM stations to act like a traditional LAN. LAN emulated over an ATM network allows you to run your existing applications basically unchanged, while the computers on which your applications are running are connected to the ATM network.

Digital implements only the LAN emulation client (LEC) and does not implement the LAN emulation server (LES), or the Broadcast and Unknown server (BUS). The LES and BUS servers must be provided by some other facility such as Digital's GIGAswitch/ATM.

Table 3–2 shows the characteristics of LAN emulation over ATM media.

Table 3–2 LAN Media Characteristics

Characteristics	Description
Speed	155 Mb/s
Max. frame size	1516, 4544, and 9234
Max. stations	LAN server determines number of stations supported
Max. LAN size	Unrestricted

3.1.2 Classical IP Over an ATM Network

OpenVMS Alpha Version 7.1 implements the standards defined in RFC 1577 (Classical IP over ATM). This implementation introduces the necessary system interface for Classical IP (CLIP) over an ATM network, and enables IP and Address Resolution Protocols (ARPs) in an ATM network environment to be configured as a logical IP subnetwork (LIS). Consult your IP product documentation for the status of support of IP over this interface.

Section 3.1.3 describes the LANCP commands to manage LAN emulation over an ATM network. Section 3.1.4 describes the commands to manage CLIP over an ATM network.

3.1.3 Managing LAN Emulation Over an ATM Network with LANCP Commands

LAN emulation over an ATM network is managed by using special LANCP command qualifiers. These qualifiers are presented in the following sections.

3.1.3.1 DEFINE DEVICE Qualifiers for LAN Emulation

Table 3–3 shows the LAN emulation qualifiers for the DEFINE DEVICE command.

Table 3–3 DEFINE DEVICE Command Qualifiers for LAN Emulation

Qualifier	Description
/ATMADDRESS	On Alpha systems, defines the LAN emulation server (LES) address for asynchronous transfer mode (ATM). Usually the address is not user specified, and this qualifier is used only if you want a specific address. By default, the address is determined by software from the configuration server for the LES. The /ATMADDRESS qualifier's syntax is as follows: `DEFINE DEVICE/ATMADDRESS=([NO]LES=the ATM address)`
/DLL	Same as the /MOPDLL qualifier.

/ELAN	The /ELAN qualifier controls the state of the LAN emulation and has two values: enable and disable. With /ELAN=ENABLE and the keyword STARTUP, the LAN emulation is loaded when LANCP starts. With /ELAN=DISABLE, the same parameters as ENABLE can be used. The /ELAN qualifier's syntax is as follows:
	
	`DEFINE DEVICE/ELAN=(...`
	`[parent=parent device,]...`
	`[name="ELAN NAME to join",]...`
	`[size=1516]...`
	`[type=CSMACD]...`
	`[Enable,]...[Disable,]...`
	`[Enable=startup,]...[Disable=startup,]`
	`...[description = "description string,"])`

(continued on next page)

Table 3–3 (Cont.) DEFINE DEVICE Command Qualifiers for LAN Emulation

Qualifier	Description
/ELAN (continued)	The meanings for the syntax for /ELAN are as follows: • *parent* The ATM adapter device name. An example of the parent device for ATMWORKS 350 is: HW*n*0, where *n* is the controller number. An example of the parent device for ATMWORKS 750 is: HC*n*0, where *n* is the controller number. • *name* Optionally specified if you want to join a specific ELAN. The default is null. • *size* Maximum frame size of the LAN you want to join. Valid sizes are 1516, 4544, or 9234 bytes. • *type* Support currently only for CSMACD. • *description* A method of describing the ELAN for display purposes only. Keywords and their meanings for /ELAN are as follows: • *Enable* Begins a join on a specified emulated LAN. It also loads the driver, if not already loaded. • *Disable* Causes a client to leave the emulated LAN.

(continued on next page)

Table 3–3 (Cont.) DEFINE DEVICE Command Qualifiers for LAN Emulation

Qualifier	Description
/MOPDLL=(*option*)	Provides the MOP downline load service settings for the device. With this qualifier, you can specify: • *enable-option* Indicates that MOP downline load service should be enabled or disabled for the device. • *exclusive-option* Indicates that no other provider of MOP downline load service is allowed on the specified LAN device at the same time as LANACP. • *knownclientsonly-option* Indicates that MOP downline load requests should be serviced only for clients defined in the LAN volatile node database. • *size-option* Specifies the size in bytes of the file data portion of each downline load message.

For more detailed information, see the *OpenVMS System Manager's Manual: Tuning, Monitoring, and Complex Systems* and the *OpenVMS System Management Utilities Reference Manual: A–L.*

3.1.3.2 LIST DEVICE Qualifiers for LAN Emulation

Table 3–4 shows the qualifiers for the LANCP LIST DEVICE command that support LAN emulation.

Table 3–4 LIST DEVICE Command Qualifiers for LAN Emulation

Qualifier	Description
/ALL	Lists all devices that match device names.
/CHARACTERISTICS	Same as the /PARAMETERS qualifier.
/DLL	Same as the DEFINE DEVICE/MOPDLL qualifier. Lists downline load characteristics.
/PARAMETERS	Displays status and related information about the device.

For more detailed information, see the *OpenVMS System Manager's Manual: Tuning, Monitoring, and Complex Systems* and the *OpenVMS System Management Utilities Reference Manual: A–L.*

3.1.3.3 SET DEVICE Qualifiers for LAN Emulation

Table 3–5 shows the qualifiers for the LANCP SET DEVICE command that support LAN emulation.

Table 3–5 SET DEVICE Command Qualifiers for LAN Emulation

Qualifier	Description
/ATMADDRESS	On Alpha systems, sets the LAN emulation server (LES) address for asynchronous transfer mode (ATM). Usually the address is not user specified, and this qualifier is used only if you want a specific address. By default the address is determined by software from the configuration server for the LES. The /ATMADDRESS qualifier's syntax is as follows: `SET DEVICE/ATMADDRESS=([NO]LES=the ATM server)`
/DLL	Same as the DEFINE DEVICE/MOPDLL qualifier.

/ELAN	Sets LAN emulation parameters. The /ELAN qualifier's syntax is as follows:
	
	`SET DEVICE/ELAN=(...`
	`[parent=parent device,]...`
	`[name="ELAN NAME to join",]...`
	`[size=1516]...`
	`[type=CSMACD]...[Create,]`
	`[Enable,]...[Disable,]...`
	`[description = "description string,"])`
	
	The meanings for the syntax for /ELAN are as follows:
	
	• *parent*
	The ATM adapter device name. An example of the parent device for ATMWORKS 350 is: HW*n*0, where *n* is the controller number. An example of the parent device for ATMWORKS 750 is: HC*n*0, where *n* is the controller number.
	
	• *name*
	Optionally specified if you want to join a specific ELAN. The default is null.
	
	• *size*
	Maximum frame size of the LAN you want to join. Valid sizes are 1516, 4544, or 9234 bytes.
	
	• *type*
	Support currently only for CSMACD.
	
	• *description*
	A method of describing the ELAN for display purposes only.

(continued on next page)

Table 3–5 (Cont.) SET DEVICE Command Qualifiers for LAN Emulation

Qualifier	Description
/ELAN (continued)	Keywords and their meanings for /ELAN are as follows: • *Create* Loads the emulation driver SYS$ELDRIVER.EXE, but does not start it. • *Enable* Begins a join on a specified emulated LAN. It also loads the driver, if not already loaded. • *Disable* Causes a client to leave the emulated LAN.
/PERMANENT	Reads the permanent database and creates device entries in the volatile database.

For more detailed information, see the *OpenVMS System Manager's Manual: Tuning, Monitoring, and Complex Systems* and the *OpenVMS System Management Utilities Reference Manual: A–L.*

3.1.3.4 SHOW DEVICE Qualifiers for LAN Emulation

Table 3–6 shows the LAN emulation qualifiers for the SHOW DEVICE command.

Table 3–6 SHOW DEVICE Command Qualifiers for LAN Emulation

Qualifier	Description
/ALL	Shows all devices that match device names.
/CHARACTERISTICS	Same as the /PARAMETERS qualifier.
/DLL	Same as the DEFINE DEVICE/MOPDLL qualifier. It shows downline load characteristics.
/PARAMETERS	Displays status and related information about the device.

For more detailed information, see the *OpenVMS System Manager's Manual: Tuning, Monitoring, and Complex Systems* and the *OpenVMS System Management Utilities Reference Manual: A–L.*

3.1.4 Managing Classical IP Over an ATM Network with LANCP Commands

Classical IP (CLIP) over an ATM network is managed by using special LANCP command qualifiers. It is important to note that the /ATMADDRESS qualifier used with the LANCP DEFINE DEVICE and SET DEVICE commands needs to be manually set. This means the address resolution protocol (ARP) server must be set up before any clients are set up. There is only one ARP server for each logical IP subnet (LIS). A node that is set up as the ARP server automatically is set up as a client.

Also note that to determine the ATM address of the ARP server, the LANCP SHOW DEVICE command with the /CHARACTERISTICS qualifier needs to be invoked. As soon as the ATM ARP server is set up, the clients on other systems may be set up by specifying the ATM address of the ARP server with the LANCP SET DEVICE command.

The LANCP qualifiers are presented in the following sections.

3.1.4.1 DEFINE DEVICE Qualifiers for CLIP

Table 3–7 shows the CLIP qualifiers for the DEFINE DEVICE command.

Table 3–7 DEFINE DEVICE Command Qualifiers for CLIP

Qualifier	Description
/ATMADDRESS	On Alpha systems, defines the LAN address resolution protocol (ARP) server address for ATM. This qualifier is required before a logical IP subnet (LIS) is enabled if the local host is not the ARP server. The /ATMADDRESS qualifier's syntax is as follows: `DEFINE DEVICE/ATMADDRESS=(ARP=atm_arp_server)`

/CLIP	On Alpha systems, the /CLIP qualifier defines a device with the TCP/IP protocol. This enables the device to see a network as being configured as one or more logical IP subnets (LIS) and allows it to transmit Ethernet frames over the ATM network. The /CLIP=ENABLE command causes the system to join the logical IP subnet when LANCP initializes. The /CLIP=DISABLE command causes the client to leave the logical IP subnet.
	
	Note that a logical IP subnet requires a server, and there must be only one server for each subnet. Communication between subnets can only be performed by a router. Also, there can only be one client for each ATM adapter.
	
	The /CLIP qualifier's syntax with standard Internet dotted notation is as follows:
	
	`DEFINE DEVICE/CLIP=(ip_subnet=a.b.c.d,`
	`ip_addr=a.b.c.d,`
	`parent=devnam,`
	`name="ip subnet name",`
	`enable, disable`
	`type=client|server)`

(continued on next page)

Table 3–7 (Cont.) DEFINE DEVICE Command Qualifiers for CLIP

Qualifier	Description
/CLIP (continued)	The meanings for the syntax of keyword and subkeyword for /CLIP are as follows: • *Enable* Joins the logical IP subnet. • *Disable* Causes a client to leave the logical IP subnet. • *ip_address=* Specifies the IP address of the CLIP client. • *parent=* Specifies the ATM adapter device name. An example of the parent device for ATMworks 350 is: HW*n*0, where *n* is the controller number. An example of the parent device for ATMworks 750 is: HC*n*0, where *n* is the controller number. • *ip_subnet=* Specifies the subnet mask of the CLIP client. • *name=* Specifies a name for the LIS to aid in operations and diagnostics. • *type=client* Starts up a CLIP client only. This is the default. The server needs to be started and the /ATMADDRESS needs to be supplied. • *type=server* Starts up a CLIP server and client. Only one server for each LIS is allowed, and the server needs to be started first.

For more detailed information, see the *OpenVMS System Manager's Manual: Tuning, Monitoring, and Complex Systems* and the *OpenVMS System Management Utilities Reference Manual: A–L.*

3.1.4.2 LIST DEVICE Qualifiers for CLIP

Table 3–8 shows the qualifiers for the LANCP LIST DEVICE command that support CLIP.

Table 3–8 LIST DEVICE Command Qualifiers for CLIP

Qualifier	Description
/ALL	Lists all devices that match device names.
/CHARACTERISTICS	Same as the /PARAMETERS qualifier.
/PARAMETERS	Displays status and related information about the device.

For more detailed information, see the *OpenVMS System Manager's Manual: Tuning, Monitoring, and Complex Systems* and the *OpenVMS System Management Utilities Reference Manual: A–L.*

3.1.4.3 SET DEVICE Qualifiers for CLIP

Table 3–9 shows the qualifiers for the LANCP SET DEVICE command that support CLIP.

Table 3–9 SET DEVICE Command Qualifiers for CLIP

Qualifier	Description
/ATMADDRESS	On Alpha systems, sets the address resolution protocol (ARP) server address for ATM. This qualifier is required before a LIS is enabled if the local host is not the ARP server. The /ATMADDRESS qualifier's syntax is as follows: `SET DEVICE/ATMADDRESS=(ARP=atm_arp_server)`

(continued on next page)

Table 3–9 (Cont.) SET DEVICE Command Qualifiers for CLIP

Qualifier	Description

/CLIP	On Alpha systems, the /CLIP qualifier sets a device with the TCP/IP protocol. This enables the device to see a network as being configured as one or more logical IP subnets (LIS) and allows it to transmit Ethernet frames over the ATM network. The /CLIP=ENABLE command causes the system to join the logical IP subnet when LANCP initializes. The /CLIP=DISABLE command causes the client to leave the logical IP subnet.
	
	Note that a logical IP subnet requires a server, and there must be only one server for each subnet. Communication between subnets can only be performed by a router. Also, there can only be one client for each ATM adapter.
	
	The /CLIP qualifier's syntax with standard Internet dotted notation is as follows:
	
	`SET DEVICE/CLIP =(create,ip_subnet=a.b.c.d,`
	`ip_addr=a.b.c.d,`
	`parent=devnam,`
	`name="ip subnet name",`
	`enable, disable`
	`type=client|server)`
	
	The meanings for the syntax of keyword and subkeyword for /CLIP are as follows:
	
	• *Create*
	Loads the CLIP driver but does not start it.
	
	• *Enable*
	Joins the logical IP subnet.
	
	• *Disable*
	Causes the node to leave the logical IP subnet.
	
	• *ip_address=*
	Specifies the IP address of the CLIP client.
	
	• *parent=*
	Specifies the ATM adapter device name. An example of the parent device for ATMworks 350 is: HW*n*0, where *n* is the controller number. An example of the parent device for ATMworks 750 is: HC*n*0, where *n* is the controller number.

(continued on next page)

Table 3–9 (Cont.) SET DEVICE Command Qualifiers for CLIP

Qualifier	Description
/CLIP (continued)	
	• *ip_subnet=* Specifies the subnet mask of the CLIP client. • *name=* Specifies a name for the LIS to aid in operations and diagnostics. • *type=client* Starts up a CLIP client only. This is the default. The server needs to be started and the /ATMADDRESS needs to be supplied. • *type=server* Starts up a CLIP server and client. Only one node for each LIS is allowed, and the server needs to be started first.
/PERMANENT	Reads the permanent database and creates device entries in the volatile database.

For more detailed information, see the *OpenVMS System Manager's Manual: Tuning, Monitoring, and Complex Systems* and the *OpenVMS System Management Utilities Reference Manual: A–L.*

3.1.4.4 SHOW DEVICE Qualifiers for CLIP

Table 3–10 shows the CLIP qualifiers for the SHOW DEVICE command.

Table 3–10 SHOW DEVICE Command Qualifiers for CLIP

Qualifier	Description
/ALL	Shows all devices that match device names.
/CHARACTERISTICS	Same as the /PARAMETERS qualifier.
/PARAMETERS	Displays status and related information about the device.

For more detailed information, see the *OpenVMS System Manager's Manual: Tuning, Monitoring, and Complex Systems* and the *OpenVMS System Management Utilities Reference Manual: A–L.*

3.2 Backup Utility: /[NO]UNSHELVE Qualifier

The /[NO]UNSHELVE qualifier controls how the Backup utility designates files from a BACKUP save operation: shelved or unshelved. File shelving migrates specific file data to offline storage transparently — the data is still available and appears as if it were still online. This allows you to easily bring the file data back online, when necessary.

The /UNSHELVE qualifier specifies that the Backup utility designate files from a BACKUP save operation as *unshelved*, which means the data is available in an online state. A *shelved* file contains no actual file data. However, the file header remains intact and online. The default behavior is unshelved except during BACKUP save operations that require the /PHYSICAL or /IMAGE qualifier. For those operations, the files remain in their original file shelving state.

For more information about file shelving, refer to the POLYCENTER Hierarchical Storage Management (HSM) for OpenVMS documentation.

3.3 DECamds: Right to Use Provided with OpenVMS License

With OpenVMS Version 7.1, the right to use DECamds is provided with the OpenVMS operating system base license. Previously, the right to use DECamds was provided with the OpenVMS Cluster Software license. Now you can use all the system management capabilities provided by DECamds in a non-clustered environment.

DECamds is a separately installable, realtime, high-performance, multisystem monitoring utility. DECamds provides system resource monitoring, investigation aid, and correction capability. This enables system managers to proactively and effectively manage multiple systems from a centralized DECwindows display.

The DECamds kit can be installed on systems running OpenVMS Version 6.1 and later. For information on how to use DECamds to monitor your system resources, see the *DECamds User's Guide*.

3.4 Dump File Features

The *OpenVMS System Manager's Manual* contains additional information about each new feature in this section.

3.4.1 Dump File Off the System Disk (DOSD)

You can now write the system dump file to a device other than the system disk. This feature is especially useful in large memory systems and in clusters with common system disks where sufficient disk space, on one disk, is not always available to support dump file requirements.

DOSD requirements are somewhat different on VAX and Alpha systems. On both systems, however, you must correctly enable the DUMPSTYLE system parameter to enable the bugcheck code to write the system dump file to an alternate device.

3.4.2 Error Log Dump File (Alpha Only)

On Alpha systems, the contents of memory-resident error log buffers are now written not only to the system dump file but also to the new error log dump file. When an operator-initiated shutdown occurs, the contents of error log buffers are written only to the error log dump file, not to the system dump file. This means that when an operator initiates shutdown, the last system crash dump is not overwritten.

The error log dump file, SYS$SPECIFIC:[SYSEXE]SYS$ERRLOG.DMP, is not provided on the distribution kit; AUTOGEN creates this file during installation. AUTOGEN also automatically determines the appropriate size of the error log dump file for your configuration and system parameters. The error log dump file must always be on the system disk.

3.4.3 Selective System Dumps: Fine-Tuning the Order in Which Processes Are Written (Alpha Only)

On Alpha systems, a new category of processes, key processes, are dumped immediately following PT (page table) space, S0/S1 space, and S2 space. These dumps also include transition pages that link back to the key process. The system manager can designate additional processes to be treated as key processes, have priority over other processes in a dump. Designating key processes ensures that any critical information provided by these processes is successfully written to the dump file, even when it is too small to contain information about all processes.

See the *OpenVMS System Manager's Manual: Tuning, Monitoring, and Complex Systems* for instructions on how to designate the order of processes within a system dump.

3.5 External Authentication Support

OpenVMS Version 7.1 provides support for external authentication. The PATHWORKS authentication module is supported as an external authenticator, providing LAN Manager authentication of OpenVMS users. (PATHWORKS for OpenVMS Version 5.0D is required.) This is expected to be the first of several authentication modules which will be available in future releases of OpenVMS.

External authentication with LAN Manager allows users to log in at the OpenVMS login prompt using their LAN Manager user IDs and passwords. When successfully authenticated, the LAN Manager user ID is mapped to the appropriate OpenVMS user name and the correct user profile is obtained.

Before users can use external authentication, it must be enabled on the system and the user accounts in the system user authorization file (SYSUAF) must be marked for external authentication. These tasks are discussed in the following section.

3.5.1 Enabling External Authentication

External authentication is controlled at two levels:

- At the system level, external authentication is enabled by defining the SYS$SINGLE_SIGNON logical name as 1 (bit 0) in SYSTARTUP_VMS.COM during system startup after all dependent products (such as PATHWORKS) have been started. (SYS$SINGLE_SIGNON is defined as a hexadecimal string bit mask.)

  An ACME module is an interface to the external authenticator. Specify the external authenticator (in this case, PATHWORKS) by defining the SYS$ACME_MODULE logical name and installing it as a shareable image. (Obtain PWRK$ACME_MODULE.EXE from the OpenVMS Version 7.1 web site, http://www.service.digital.com/openvms.)

  For example:

```
$ DEFINE/SYSTEM/EXECUTIVE SYS$ACME_MODULE SYS$SHARE:PWRK$ACME_MODULE.EXE
$ INSTALL ADD SYS$ACME_MODULE /OPEN/HEAD/SHARE
$ DEFINE/SYSTEM/EXECUTIVE SYS$SINGLE_SIGNON 1
```

  If the SYS$SINGLE_SIGNON logical name is not set, external authentication is disabled regardless of any other settings. (See the Managing System Access chapter in the *OpenVMS Guide to System Security* for more information on the SYS$SINGLE_SIGNON logical name bits.)

- At the user level, external authentication is enabled by a new flag, EXTAUTH, in the SYSUAF record. When set, the EXTAUTH flag denotes that the user is to be externally authenticated. For example, in the Authorize utility, enter:

  ```
  ADD username /FLAG=([NO]EXTAUTH)
  MODIFY username /FLAG=([NO]EXTAUTH)
  ```

  (See the *OpenVMS System Management Utilities Reference Manual* for more information on the Authorize utility EXTAUTH flag, and the *OpenVMS System Services Reference Manual: GETQUI–Z* for more information on the UAI$V_EXTAUTH bit in the SYS$GETUAI and SYS$SETUAI system services UAI$_FLAGS item code.)

By default, external authentication is disabled at both the system and user levels.

3.5.2 External Authentication Over a Network

External authentication cannot occur if the network connection required for the authentication is down. However, privileged users can enter the /LOCAL_PASSWORD qualifier after their OpenVMS user name at the login prompt to inform OpenVMS that local authentication is to be performed. Users should specify their OpenVMS user name and password when using the /LOCAL_PASSWORD qualifier.

Because the use of the /LOCAL_PASSWORD qualifier is effectively overriding the security policy established by the system manager, it is only allowed under the following conditions:

- When the account being logged in to has SYSPRV as an authorized privilege. This allows the system manager to gain access to the system when the network is down.
- When bit 1 is set in the SYS$SINGLE_SIGNON logical name, which allows nonprivileged users (who are normally externally authenticated) to log in locally.

For more information on the /LOCAL_PASSWORD qualifier to LOGINOUT, see the *OpenVMS Utility Routines Manual*.

3.5.3 Setting a New Password

If you are an externally authenticated user, the DCL command SET PASSWORD sends the password change request to the external authenticator and also changes your password on your OpenVMS system.

A system manager can set an externally authenticated user's password by using a utility provided by the external authenticator. In the case of LAN Manager, PATHWORKS provides a NET PASSWORD command. Using this method, the new password is propagated to the network immediately.

3.5.4 How External Authentication Works

External authentication allows a user who is logging in to OpenVMS to be authenticated by an external entity (such as the PATHWORKS authentication module). OpenVMS does not check the SYSUAF to validate the user's password.

3.5.4.1 User Name Mapping and Password Verification

To be authenticated using LAN Manager user IDs and passwords, an externally authenticated user provides his or her LAN Manager user ID and password at the OpenVMS login prompt. The user IDs and passwords are used for authentication by a LAN Manager domain controller. When performing user name mapping, OpenVMS first tries to locate a match in the SYSUAF and uses that name if it finds a match; otherwise, it searches the LAN Manager database for a matching user ID. When successfully authenticated, the LAN Manager user ID is mapped to the appropriate OpenVMS user name to obtain the correct user profile, and the login sequence is completed.

External authentication is supported for interactive logins (including DECwindows) and network logins where a proxy is used or user ID/password is supplied.

Externally authenticated users are considered to have a single password and are not subject to normal OpenVMS password policy (password expiration, password history, minimum and maximum password length restrictions), but are instead subject to any defined LAN Manager policy. All other OpenVMS account restrictions remain in effect, such as disabled accounts, modal time restrictions, quotas, and so on.

Externally authenticated users are identified by having the EXTAUTH flag set in their SYSUAF record. OpenVMS users whose accounts do not have the EXTAUTH flag set are not affected by external authentication.

3.5.4.2 Password Synchronization

Although passwords for externally authenticated users are verified using the LAN Manager database, OpenVMS attempts to keep the LAN Manager and SYSUAF password fields synchronized. This synchronization occurs from LAN Manager to the SYSUAF.

Password synchronization is enabled by default.

Synchronization takes place at the completion of a successful externally authenticated login. If the LAN Manager password is different than the password stored in the SYSUAF file, LOGINOUT attempts to update the SYSUAF password field with the LAN Manager password. (Synchronization may not be possible due to the different sets of valid characters allowed by OpenVMS and LAN Manager.)

If required, the password synchronization feature can be selectively turned off. (See the Managing System Access chapter in the *OpenVMS Guide to System Security* for more information on the SYS$SINGLE_SIGNON logical name bits, which control the enabling and disabling of password synchronization.)

For more detailed information, see the *OpenVMS Guide to System Security*.

3.6 Fast Ethernet LANs (Alpha Only)

Fast Ethernet (802.3u), also known as 100BaseT, is an extension of the regular Ethernet 10BaseT standard. Fast Ethernet raises the data transmission capacity of 10BaseT from 10Mb/s to 100Mb/s while retaining Carrier Sense Multiple Access with Collision Detection (CSMA/CD) transmission protocol of 10BaseT Ethernet. Fast Ethernet works with several different types of cable including basic twisted-pair wiring. Because Fast Ethernet is a continuation of the former Ethernet standard, the same procedures, network analysis tools, and applications that run with the 10BaseT Ethernet network work with the 100BaseT network.

Table 3–11 shows the types of cabling that Fast Ethernet uses.

Table 3–11 Fast Ethernet Cabling

Cable	Description
100BaseTX	Works with twisted-pair cabling standards. It provides full-duplex performance with network servers, using only two of the four pairs of wires.
100BaseT4	Works with twisted-pair cabling standards. It uses four pairs of wiring with one pair for transmission, another for reception, and two pairs that can be used to either transmit or receive data. It does not support full-duplex operations.
100BaseFX	Uses fiber optic cabling mainly for backbones by connecting Fast Ethernet repeaters placed around a building. It gives protection from electromagnetic noise and increases security. It also allows longer distances between network devices.

The end-to-end diameter or segment diameter of a network for Fast Ethernet is limited to 250 meters. The 10Mb/s Ethernet network has a limit of 2,500 meters. Fast Ethernet adheres to the 10BaseT rule that a computer can be no more than 100 meters from a hub.

OpenVMS Version 7.1 supports the following two adapters:

DE500-XA
DE500-AA

Each of these adapters may be run at 10Mb/s or 100Mb/s. The 10Mb/s or 100Mb/s transmission rate is determined when the driver is started.

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

3.7 High SCSI IDs on Wide Adapters (Alpha Only)

OpenVMS Alpha Version 7.1 device driver support of high SCSI IDs allows host adapters that support the wide SCSI bus to configure and use up to 16 devices per SCSI bus.

KZTSA, KZPSA, and the QLogic wide adapters (including KFTIA fast wide differential ports, KZPDA, KZPSM, and the AlphaStation 600 embedded SCSI port) currently support high SCSI IDs.

Note that although the maximum allowable number of devices per SCSI bus has increased, the SCSI bus length requirements have not. When configuring devices above the previous limit of 8 devices, failure to adhere to bus length requirements can produce unpredictable results.

Supported SCSI bus lengths are as follows:

- 3 meters (10 feet) on fast single-ended buses
- 6 meters (20 feet) on slow single-ended buses
- 25 meters on differential buses

To configure wide IDs on a BA356 box, refer to the BA356 manual *StorageWorks Solutions BA356-SB 16-Bit Shelf User's Guide*.

If you mix narrow and wide devices on a wide bus, make sure that the hosts and all narrow devices have IDs in the 0 to 7 range. Do not install narrow devices in a storage shelf that is configured for IDs in the range of 8 to 15.

3.8 LAN Control Program (LANCP) Utility: New Commands

OpenVMS Version 7.1 supports the following new LANCP utility commands:

- CONVERT DEVICE_DATABASE
- CONVERT NODE_DATABASE

These commands convert earlier LANCP database formats to the current Version 7.1 format. Note that although both LANCP and LANACP can read earlier database formats, entries can only be written in the Version 7.1 format.

3.8.1 CONVERT DEVICE_DATABASE

The CONVERT DEVICE_DATABASE command converts an OpenVMS Version 6.2 format LANCP device database to an OpenVMS Version 7.1 format device database. This command requires SYSPRV privilege. This command has no parameters or qualifiers.

For more information, see the *OpenVMS System Management Utilities Reference Manual: A–L*.

3.8.2 CONVERT NODE_DATABASE

The CONVERT NODE_DATABASE command converts an OpenVMS Version 6.2 format LANCP node database to an OpenVMS Version 7.1 format node database. This command requires SYSPRV privilege. This command has no parameters or qualifiers.

For more information, see the *OpenVMS System Management Utilities Reference Manual: A–L*.

3.9 License Management Facility (LMF) Features

The OpenVMS License Management Facility (LMF) offers the following new features:

- Ability to load more than one type of license for a product on the same node
- Use of reservation lists to set aside allocation units for specified users (a license with a reservation list cannot be combined with a license without a reservation list)
- Ability to modify the selection weight, which determines the grant order in which LMF checks multiple licenses prior to loading

Use the License Management utility (LICENSE), the command-line interface for managing software licenses, to implement these new features.

Table 3–12 shows the LICENSE qualifiers and options new for this release of the OpenVMS operating system.

Table 3–12 LICENSE Enhancements

Command	Qualifier	Description
LICENSE LIST	/SELECTION_WEIGHT	Displays the current selection weight of a PAK.
LICENSE LOAD	/UNLOAD	Default. When requested to load a license that is currently loaded, LMF first automatically unloads it and then loads the latest license.
	/NOUNLOAD	LMF does not unload the currently loaded license before loading the latest license.
LICENSE MODIFY	/COMBINE	Adds the /COMBINE option to a PAK. If PAKs can be combined, LMF combines them during license loading.
	/NOCOMBINE	Removes the /COMBINE option from a PAK.
	/SELECTION_WEIGHT=*n*	Modifies the current selection weight of a PAK.
LICENSE REGISTER	/OPTIONS=VAX_ALPHA	Registers an Availability license that is valid for both OpenVMS VAX and OpenVMS Alpha systems.
	/OPTIONS=USER	Designates a User license.

For detailed information, see the *OpenVMS License Management Utility Manual*.

In addition, the displays of the SHOW LICENSE command are updated to show any new options that you enter. For more information, see the *OpenVMS DCL Dictionary*.

3.10 Networking Support

OpenVMS Version 7.1 provides you with the flexibility to choose the network protocol of your choice. Whether you require TCP/IP, OSI, or DECnet, OpenVMS enables you to simply choose the protocol or combination of protocols that work best for your network.

If you are installing OpenVMS Alpha Version 7.1, you will be able to easily choose from among the following network options now integrated into the operating system installation procedure:

- Digital TCP/IP Services for OpenVMS
- DECnet-Plus (formerly called DECnet/OSI)
- No Network

If you are upgrading to DECnet-Plus from an earlier version of OpenVMS, the network upgrade is performed automatically as part of the operating system upgrade procedure.

Note

DECnet-Plus has replaced DECnet for OpenVMS Phase IV in both the installation and upgrade procedures for OpenVMS. To install or upgrade to DECnet Phase IV, follow the instructions in the OpenVMS *Upgrade and Installation Manual* on how to install DECnet Phase IV as a layered product.

3.10.1 DECnet-Plus for OpenVMS

With DECnet-Plus for OpenVMS Version 7.1 software, OpenVMS systems can communicate with each other and with systems produced by other vendors. DECnet-Plus provides true network independence: you get the full functionality of DECnet Phase IV, additional DECnet enhancements, and the ability to use DECnet over a variety of network protocols, such as Phase IV, TCP/IP, and OSI.

All OpenVMS Version 7.1 customers will receive a complimentary DECnet-Plus Starter Kit, which includes media and selected documentation. The starter kit documentation received will be in either online or printed format depending on the OpenVMS offering purchased. In future versions of OpenVMS, customers who do not receive the OpenVMS CD–ROM must purchase the media and documentation for Digital's networking products (such as DECnet Phase IV, DECnet-Plus, and Digital TCP/IP Services for OpenVMS) separately.

For complete information about the benefits of using DECnet-Plus software, see the *DECnet-Plus Introduction and User's Guide*.

3.10.1.1 New Features of DECnet-Plus

The new features of DECnet-Plus for OpenVMS introduced in Version 7.1 are:

- Host-based routing
- Fast configuration

Host-based routing allows an OpenVMS system to operate as a DECnet-Plus intermediate system (IS) in a routing vector domain. This feature is useful for those configurations where you need to route from a local area network (LAN) to a wide area network (WAN) and want to use an existing system to do the routing rather than investing in a dedicated router. Host-based routing is not intended to be used in network configurations that have high-throughput requirements.

For upgrades from DECnet Phase IV to DECnet-Plus, the fast-configuration option allows system or network managers to use the NET$CONFIGURE.COM procedure to configure DECnet-Plus quickly.

For complete information about the DECnet-Plus new features and functionality, see the *DECnet-Plus for OpenVMS Release Notes*.

3.10.1.2 Upgrading to DECnet-Plus

You can upgrade a network node or system from DECnet Phase IV to DECnet-Plus software in less than 1 hour. The Network Command Protocol (NCP) emulator makes the upgrade easy without disrupting your network's day-to-day operations.

Upgrading to DECnet-Plus brings significant enhancements to your DECnet network as well as global communication between independent networks. The software allows DECnet Phase IV applications to run over DECnet-Plus and TCP/IP without modification.

For DECnet-Plus upgrade and installation information, see the *DECnet-Plus Installation and Basic Configuration* manual.

3.10.2 Digital TCP/IP Services for OpenVMS

Digital TCP/IP Services for OpenVMS provides you with the following features as part of the Version 4.1 product:

- Support for the Post Office Protocol (POP), Version 3 (RFC 1725)

  The new UCX POP server ensures that mail is always accepted.

- Enhanced network management through the SNMP Extensible Agent (eSNMP) software

  With eSNMP, you can create subagents that manage other entities on your network, such as hosts, routers, and terminal servers.

- Support for the Finger utility

  The Finger utility enables you to retrieve and display information about users on a particular host system.

- File display capability

  The new FTP command VIEW lets you display the contents of a file on your current output device.

- Flexible file naming

  Using the UCX management command ADD EXPORT/OPTIONS=NAME_ CONVERSION, you can now create files and directories in an OpenVMS file system using names that do not conform to the standard OpenVMS file-naming rules.

- Support for XQP+

  The NFS server now provides access to XQP+, the eXtended QIO processor. You can configure the NFS server to take advantage of the performance enhancements offered by XQP+, such as improved file system concurrency and XQP+ threads.

For detailed information about these features, see the *Digital TCP/IP Services for OpenVMS Release Notes*.

3.11 OpenVMS Cluster Features

The following OpenVMS Cluster features are included in OpenVMS Version 7.1.

- New MEMORY CHANNEL high-performance interconnect for PCI-based Alpha systems that offers significant increases in throughput and a decrease in latency associated with traditional I/O processing
- New CI-to-PCI adapter (CIPCA) that enables PCI Alpha servers to connect to the CI
- New LAN$POPULATE support for migrating satellite booting
- Lock manager enhancements to support large database and server applications
- New CLUSTER_CREDITS system parameter to support large, lock-intensive applications such as large scale databases
- New options (Make Root and Delete Root) in CLUSTER_CONFIG.COM and CLUSTER_CONFIG_LAN.COM
- New adapter support for SCSI OpenVMS Cluster configurations
- New SCSI device naming that alleviates device naming conflicts
- SCSI wide adapter support for up to 16 SCSI devices per SCSI bus
- New OpenVMS Cluster Compatibility Kit to provide selected OpenVMS Version 7.1 features and quality improvements for OpenVMS Version 6.2 systems

These features further enhance the performance, availability, and functionality of OpenVMS Clusters. The following sections describe these features in more detail.

3.11.1 MEMORY CHANNEL Interconnect

MEMORY CHANNEL is a new, high-performance cluster interconnect for PCI-based Alpha systems. MEMORY CHANNEL provides node-to-node communications and is used with one or more other interconnects which provide storage and network communications.

MEMORY CHANNEL enables a system to write data very quickly to the memory of other systems, delivering up to 100 Mb/s aggregate bandwidth. MEMORY CHANNEL supports a maximum of 4 nodes in a 10 foot radial topology.

MEMORY CHANNEL improves throughput in high-performance databases and other applications that generate heavy OpenVMS Lock Manager traffic. With the current MEMORY CHANNEL adapters, the achievable Lock Manager performance with MEMORY CHANNEL is approximately two to three times that of a CI (depending on CPU type). Note that acheiving such performance will consume approximately three to four times the compute overhead of a CI.

MEMORY CHANNEL can be used to offload internode cluster traffic (such as lock management communication) from existing interconnects—CI, DSSI, FDDI, and Ethernet—so that they can process storage and network traffic more effectively.

For a detailed description of MEMORY CHANNEL, see *Guidelines for OpenVMS Cluster Configurations*.

3.11.2 New CI-to-PCI Adapter, CIPCA

With the release of OpenVMS Alpha Version 6.2–1H2, Digital introduced a CI-to-PCI adapter (CIPCA) that allows PCI-based AlphaServer systems to connect directly to CI-based OpenVMS Clusters. For CI-based clusters, OpenVMS Version 7.1 supports up to 26 CIPCA adapters per system, and a combination of CIXCD and CIPCA adapters on the AlphaServer 8400.

CIPCA support for Alpha servers provides the following features and benefits to customers:

Feature	Benefit
Lower entry cost and more configuration choices	If you require midrange compute power for your business needs, CIPCA enables you to integrate midrange Alpha servers into your existing CI cluster.
High-end Alpha speed and power	If you require maximum compute power, you can use the CIPCA with both the AlphaServer 8200 systems and AlphaServer 8400 systems that have PCI and EISA I/O subsystems.
Cost-effective Alpha migration path	If you want to add Alpha servers to an existing CI VAXcluster, CIPCA provides a cost-effective way to start migrating to a mixed-architecture cluster in the price/performance range that you need.
Advantages of the CI	The CIPCA connects to the CI, which offers the following advantages: • High speed to accommodate larger processors and I/O-intensive applications. • Efficient, direct access to large amounts of storage. • Minimal CPU overhead for communication. CI adapters are intelligent interfaces that perform much of the communication work in OpenVMS Cluster systems. • High availability through redundant, independent data paths, because each CI adapter is connected with two pairs of CI cables. • Multiple access paths to disks and tapes.

For more information about CIPCA, see the *Guidelines for OpenVMS Cluster Configurations*.

3.11.3 New LAN$POPULATE Support

The LAN$POPULATE command now provides support for migrating satellite Maintenance Operations Protocol (MOP) booting from DECnet-Plus to LANCP.

LANCP is the preferred service for MOP downline loading to boot satellites in an OpenVMS Cluster. LAN$POPULATE generates command procedures that assist in migration from DECnet MOP loading to LANCP MOP loading.

Note

If you plan to use LANCP in place of DECnet, and you also plan to move from DECnet Phase IV to DECnet-Plus, Digital recommends that you do so in the following order:

1. Replace DECnet with satellite booting (MOP downline load service), using LAN$POPULATE.COM.
2. Migrate from DECnet Phase IV to DECnet-Plus.

For more information about LAN$POPULATE, see *OpenVMS Cluster Systems*.

3.11.4 Lock Manager: Increased Quotas and Limits

The OpenVMS lock manager has been enhances for OpenVMS Version 7.1. Some internal restrictions on the number of locks and resources available on the system have been eased and a method to allow enqueue limit quota (ENQLM) of greater than 32767 has been added. No application changes are required to take advantage of these increases.

Specifically, the OpenVMS lock manager includes the following additions:

- ENQLM greater than 32767 allowed

  If you set ENQLM to a value of 32767 in the SYSUAF.DAT file, OpenVMS treats it as no limit and allows an application to own up to 16,776,959 locks, the architectural maximum allowed by the OpenVMS lock manager.
- Sub-resources and sub-locks greater than 65535 allowed
- Resource hash table greater than 65535 allowed
- LOCKIDTBL size restrictions removed (LOCKITDBL_MAX obsolete)

While most processes do not require very many locks simultaneously (typically less than 100), large scale database or server applications can easily exceed the previous thresholds. For more information about these enhancements, see the *OpenVMS Programming Concepts Manual*.

3.11.5 New CLUSTER_CREDITS System Parameter

The CLUSTER_CREDITS parameter specifies the number of per-connection buffers a node allocates to receiving VMS$VAXcluster communications. This system parameter is provided to support lock-intensive applications, such as large scale-databases, which may require more per-connection buffers. Prior to this release, it was not possible to change the default setting.

This system parameter is not dynamic; that is, if you change the value, you must reboot the node on which you changed it.

A shortage of credits affects performance, since message transmissions are delayed until free credits are available. These are visible as credit waits in the SHOW CLUSTER display.

For instructions for using this system parameter, see *OpenVMS Cluster Systems*.

3.11.6 New Cluster Configuration Options

Two new options, Make Root and Delete Root, have been added to the main menu of both CLUSTER_CONFIG.COM and CLUSTER_CONFIG_LAN.COM, as shown in the following example. These options replace the MAKEROOT.COM command procedure.

```
MAIN MENU

   1. ADD an Alpha node to the cluster.
   2. REMOVE a node from the cluster.
   3. CHANGE a cluster member's characteristics.
   4. CREATE a duplicate system disk for node-name.
   5. MAKE a directory structure for a new root on a system disk.
   6. DELETE a root from a system disk.
   7. EXIT from this procedure.
```

In addition to these new options, you will now get a shortened main menu when you execute either CLUSTER_CONFIG.COM or CLUSTER_CONFIG_LAN.COM on a node that does not already belong to a cluster.

3.11.7 New SCSI Adapter Support

The KZPSA and KZTSA adapters are SCSI adapters that connect a PCI bus (KZPSA) or a TURBOchannel bus (KZTSA) to a single 16-bit fast-wide differential SCSI bus. With the Version 6.2–1H3 release of the OpenVMS operating system, you can share access to storage by configuring the KZPSA or KZTSA adapters on a multihost SCSI bus.

Note

The minimum adapter firmware revision that supports the KZPSA in a SCSI cluster is A10. The minimum adapter firmware revision that supports the KZTSA in a SCSI cluster is A10_1.

The KZPSA and KZTSA adapters are supported in multihost configurations on all Alpha computers that support them in single-host configurations. You can configure up to three SCSI host adapters on the same SCSI bus, using any combination of KZPSA, KZTSA, and KZPAA adapters. Each system can be connected with up to six shared buses. The KZPSA and KZTSA adapters use fast and wide SCSI data transfers when interacting with storage devices that support those modes.

For more information about OpenVMS multihost SCSI configurations, see *Guidelines for OpenVMS Cluster Configurations*. Refer to the OpenVMS SPD (25.01.xx) for the list of systems that support the KZPSA and KZTSA adapters in single-host configurations and for detailed configuration rules.

3.11.8 New SCSI Device Naming Scheme

Port allocation classes provide a new method for naming SCSI devices attached to Alpha systems in an OpenVMS Cluster. Devices that use the new naming scheme are also accessible on OpenVMS VAX systems via the MSCP server.

Port allocation classes are designed to solve the naming and configuration conflicts that can occur when configuring a large OpenVMS Cluster that includes multiple hosts sharing one or more storage interconnects and multiple disks, some connected to the shared interconnect and some connected to single-host SCSI systems.

Note

This optional feature is intended primarily for changing or creating a new cluster configuration that contains multiple SCSI buses.

For more information about port allocation classes, see *OpenVMS Cluster Systems*.

3.11.9 SCSI Support for 16 Devices

Wide adapters enable the use of 16 data lines for device identification. SCSI wide adapter support enables the configuration and use of up to 16 devices per SCSI bus. For more information, see Section 3.7.

3.11.10 OpenVMS Cluster Compatibility Kit

The OpenVMS Cluster Compatibility Kit is required for customers with mixed-version and mixed-architecture clusters that include systems running OpenVMS Version 7.1 and OpenVMS Version 6.2. It provides OpenVMS Version 7.1 Volume Shadowing, Mount, lock manager, and other quality improvements for Version 6.2 systems.

This kit also includes limited support for port allocation classes for Version 6.2 systems. Port allocation classes are a naming option for SCSI devices on systems running OpenVMS Alpha Version 7.1. OpenVMS Version 6.2 systems that have installed the Cluster Compatibility Kit can access SCSI disks, named with port allocation classes, that are connected to OpenVMS Alpha Version 7.1 systems.

For more information about the OpenVMS Cluster Compatibility Kit, see the *OpenVMS Version 7.1 Release Notes*.

3.11.11 Warranted and Migration Support for OpenVMS Cluster Configurations

OpenVMS Alpha Version 7.1 and OpenVMS VAX Version 7.1 provide two levels of support, warranted and migration, for mixed-version and mixed-architecture OpenVMS Cluster systems.

Warranted support means that Digital 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 Digital has qualified the versions for use together in configurations that are migrating in a staged fashion to a newer version of OpenVMS VAX or to OpenVMS Alpha. Problem reports submitted against these configurations will be answered by Digital. However, in exceptional cases Digital may request that you move to a warranted configuration as part of answering the problem.

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

Figure 3–1 shows the level of support provided for all possible version pairings.

Figure 3–1 OpenVMS Cluster Version Pairings

	Alpha Version 6.2–*xxx*	Alpha Version 7.0	Alpha Version 7.1
VAX Version 6.2–*xxx*	WARRANTED	Migration	Migration
VAX Version 7.0	Migration	WARRANTED	Migration
VAX Version 7.1	Migration	Migration	WARRANTED

ZK–7191A–GE

Note that Digital does not support the use of Version 7.1 with Version 6.1 (or earlier versions) in an OpenVMS Cluster. In many cases, mixing Version 7.1 with versions prior to Version 6.2 will successfully operate, but Digital cannot commit to resolving problems experienced with such configurations.

3.12 OpenVMS Management Station: Offers Enhanced Printer Management

OpenVMS Management Station now makes it easy for you to manage a wide range of printers and print queues across multiple OpenVMS Cluster systems and OpenVMS nodes. Plus, the printer monitoring feature allows you to detect and correct printer problems quickly and efficiently.

You no longer need to maintain complicated command files to control your printer environment. You can create, delete, and manage a printer and its related queues, as well as perform print job management for those printers, from an easy-to-use Windows interface.

Some of the tasks you can now perform include:

- Monitor one or more printers
- Examine and modify printer attributes
- Delete printers and their associated queues
- Create printers and their associated queues
- Examine and modify queue attributes
- Examine and modify job attributes, requeue jobs, delete jobs

For more information about this feature, see the *OpenVMS Management Station Overview and Release Notes*.

3.13 OpenVMS Volume Shadowing: Minimerge Capability for System Disks

OpenVMS Volume Shadowing Version 7.1 has been enhanced to permit system disk shadow sets to be minimerged. This provides much faster merge operations for shadowed system disks, eliminating the I/O overhead associated with a full merge operation. Minimerge is an MSCP feature that is available for cluster configurations with DSSI- and CI-based shadow sets. Note that non-system disks have benefited from the minimerge feature for several releases.

To take advantage of this feature, it is necessary to use another new feature in OpenVMS Version 7.1 — the ability to write system crash dump files off the system disk. When the crash dump file is written to a non-system disk, the system disk minimerge feature can be enabled.

For more information on volume shadowing and the OpenVMS Cluster Compatibility Kit, see the *OpenVMS Version 7.1 Release Notes*.

3.14 POLYCENTER Software Installation Utility: New Subcommands and Displays

The POLYCENTER Software Installation Utility PRODUCT command has the following new subcommands:

- EXTRACT FILE
- EXTRACT PDF
- EXTRACT PTF
- LIST
- SHOW UTILITY

In addition, the format and content of the display has changed for the following PRODUCT subcommands:

- FIND
- SHOW HISTORY
- SHOW OBJECT
- SHOW PRODUCT

For PRODUCT FIND, a /FULL qualifier has been added. Many improvements have also been made to the interactive dialog for several PRODUCT subcommands, especially PRODUCT INSTALL. More detailed examples are in the PRODUCT command section of the *OpenVMS System Management Utilities Reference Manual*.

Following are brief explanations of the new subcommands. More detailed explanations and examples are in the POLYCENTER section of the *OpenVMS System Management Utilities Reference Manual*.

3.14.1 EXTRACT FILE Subcommand

The EXTRACT FILE subcommand retrieves a user-specified file or files from a sequentially formatted software product kit. A file type of .PCSI denotes a sequential kit. The original name of the file is preserved when it is extracted.

3.14.2 EXTRACT PDF Subcommand

The EXTRACT PDF subcommand retrieves the product description file (PDF) from a sequentially formatted software product kit. A file type of .PCSI denotes a sequential kit. The file type of the extracted PDF file is .PCSI$DESCRIPTION.

3.14.3 EXTRACT PTF Subcommand

The EXTRACT PTF subcommand retrieves the product text file (PTF) from a sequentially formatted software product kit. A file type of .PCSI denotes a sequential kit. The PTF is stored in a product kit as a text library file. The file type of the extracted PTF file is .PCSI$TLB. In addition, a text file version of this text library file is created with a file type of .PCSI$TEXT.

3.14.4 LIST Subcommand

The LIST subcommand lists the names of the files contained in a sequentially formatted software product kit. A file type of .PCSI denotes a sequential kit. All files in a kit are listed unless you use the /SELECT qualifier to specify a subset of the files.

3.14.5 SHOW UTILITY Subcommand

The SHOW UTILITY subcommand displays the version of the POLYCENTER Software Installation utility that implements the PRODUCT command.

3.15 SYSMAN Utility: New Qualifiers

New qualifiers make several SYSMAN commands more usable by allowing you to do the following:

- Control the rate at which information is displayed
- Verify that you want to perform an operation on a particular node

The following sections describe the commands and each new qualifier.

3.15.1 DO/CONFIRM Qualifier

You can use the /CONFIRM qualifier to verify that you want to perform a DO command on each node you have specified with the SYSMAN command SET ENVIRONMENT.

Example

The following example shows how to control whether the system displays time for each node in a cluster.

```
$ MCR SYSMAN
SYSMAN> SET ENVIRONMENT/CLUSTER

%SYSMAN-I-ENV, current command environment:
        Clusterwide on local cluster
        Username KIERSTEIN   will be used on nonlocal nodes

SYSMAN> DO/CONFIRM SHOW TIME

Execute command for node EXPERT? [N]: Y Return

%SYSMAN-I-OUTPUT, command execution on node EXPERT
  22-MAR-1996 09:40:28

Execute command for node MODERN [N]: Y Return

%SYSMAN-I-OUTPUT, command execution on node MODERN
  22-MAR-1996 09:40:56

Execute command for node IMPOSE? [N]: N Return

Execute command for node ADU26A? [N]: Y Return
   .
   .
   .
```

3.15.2 DO/PAUSE Qualifier

You can use the /PAUSE qualifier to control the rate at which the system displays information. Using the /PAUSE qualifier causes the system to display information about one node at a time, prompting you to press Return when you are ready to display the information about the next node. For example:

```
SYSMAN> DO/PAUSE SHOW TIME

%SYSMAN-I-OUTPUT, command execution on node EXPERT
  22-MAR-1996 09:40:13

Press return to continue Return

%SYSMAN-I-OUTPUT, command execution on node MODERN
  22-MAR-1996 09:40:41

Press return to continue Return

%SYSMAN-I-OUTPUT, command execution on node IMPOSE
  22-MAR-1996 09:39:46

Press return to continue Return
   .
   .
   .
```

3.15.3 IO REBUILD Command (Alpha Only)

The IO REBUILD command reconstructs device configuration tables in preparation for using the IO AUTOCONFIGURE command to reconfigure the system.

You must have CMKRNL privilege to use this command.

/VERIFY Qualifier

The /VERIFY qualifier with the IO REBUILD command causes SYSMAN to read and process the files SYS$SYSTEM:SYS$USER_CONFIG.DAT and SYS$SYSTEM:CONFIG.DAT, but not to apply the files to the I/O database. Messages will be displayed for any errors that are encountered. Developers can use the IO REBUILD/VERIFY command to test new changes to SYS$SYSTEM:SYS$USER_CONFIG.DAT without modifying the current system.

3.15.4 PARAMETERS SHOW/PAUSE Qualifier

You can use the /PAUSE qualifier to control the rate at which the system displays information about parameters. Using the /PAUSE qualifier causes the system to display information about one node at a time, prompting you to press Return when you are ready to display information about the next node. For example:

```
SYSMAN> PARAMETERS SHOW/PAUSE MAXPROCESSCNT

Node EXPERT:   Parameters in use: ACTIVE
Parameter Name         Current    Default    Minimum    Maximum Unit  Dynamic
--------------         -------    -------    -------    ------- ----  -------
MAXPROCESSCNT              160         32         12       8192 Processes

Press return to continue Return

Node MODERN:   Parameters in use: ACTIVE
Parameter Name         Current    Default    Minimum    Maximum Unit  Dynamic
--------------         -------    -------    -------    ------- ----  -------
MAXPROCESSCNT              157         32         12       8192 Processes

Press return to continue Return
```

```
Node IMPOSE:   Parameters in use: ACTIVE
Parameter Name         Current    Default    Minimum    Maximum Unit  Dynamic
--------------         -------    -------    -------    ------- ----  -------
MAXPROCESSCNT               50         32         12       8192 Processes

Press return to continue Return
    .
    .
    .
```

3.16 System Parameters: New and Changed

The following sections describe the new and changed system parameters for OpenVMS Version 7.1.

3.16.1 CLUSTER_CREDITS Parameter

CLUSTER_CREDITS specifies the number of per-connection buffers a node allocates to receiving VMS$VAXcluster communications.

If the SHOW CLUSTER command displays a high number of credit waits for the VMS$VAXcluster connection to a remote node, you might consider increasing the value of CLUSTER_CREDITS on the remote node. However, in large cluster configurations, setting this value unnecessarily high will consume a large quantity of nonpaged pool. Each buffer is at least SCSMAXMSG bytes in size but might be substantially larger depending on the underlying transport.

All nodes in the cluster do not require the same value for CLUSTER_CREDITS. For small or memory-constrained systems, the default value of CLUSTER_ CREDITS should be adequate.

3.16.2 DEVICE_NAMING Parameter (Alpha Only)

DEVICE_NAMING is a bit mask indicating whether port allocation classes are used in forming SCSI device names.

Following is the bit definition:

Bit	Definition
0	If 1, enable new naming.

3.16.3 GBLPAGFIL and GBLPAGES Parameters (Alpha Only)

Note that these two system parameters are now dynamic. You do not need to reboot the system after you change their values.

3.16.4 LAN_FLAGS Parameter (Alpha Only)

LAN_FLAGS is a bit mask used to enable features in the local area networks (LAN) port drivers and support code.

Following is the bit definition:

Bit	Description
Bit 0	The default of 0 indicates that ATM devices run in SONET mode. If set to 1, this bit indicates ATM devices run in SDH mode.

Bit	Description
Bit 1	If set, this bit enables a subset of the trace and debug messages in the LAN port drivers and support code.
Bit 2	If set, this bit enables all trace and debug messages in the LAN port drivers and support code.

LAN_FLAGS is a DYNAMIC parameter.

3.16.5 MC_SERVICES Parameters

The following MEMORY CHANNEL system parameters have been added. These parameters control the node behavior in a MEMORY CHANNEL cluster. For information about the individual parameters, see the *OpenVMS Version 7.1 Release Notes*.

Parameter	Description
MC_SERVICES_P0	Controls whether other MEMORY CHANNEL nodes in the cluster continue to run if this node bugchecks or shuts down.
MC_SERVICES_P1	This parameter is reserved by Digital. Its value must be the same on all nodes connected by MEMORY CHANNEL.
MC_SERVICES_P3	Specifies the maximum number of MEMORY CHANNEL tags supported.
MC_SERVICES_P4	Specifies the maximum number of regions supported.
MC_SERVICES_P5	This parameter is reserved by Digital. Its value must be the same on all nodes connected by MEMORY CHANNEL.
MC_SERVICES_P6	Specifies MEMORY CHANNEL message size, the body of an entry in a work queue.
MC_SERVICES_P7	Specifies whether to suppress or display messages about MEMORY CHANNEL activities on this node.
MC_SERVICES_P8	This parameter is reserved by Digital. Its value must be the same on all nodes connected by MEMORY CHANNEL.
MC_SERVICES_P9	Specifies the number of initial entries in a single channel's free queue.

3.16.6 MSCP_CMD_TMO Parameter

MSCP_CMD_TMO is the time in seconds that the OpenVMS MSCP server uses to detect MSCP command timeouts. The default value for MSCP_CMD_TMO is 600.

If command timeout errors are being logged on client nodes, setting this parameter to a nonzero value on OpenVMS servers reduces the number of errors logged. Increasing the value of this parameter reduces the number of client MSCP command timeouts and increases the time it takes to detect faulty devices.

If you need to decrease the number of command timeout errors, Digital recommends you set an initial value of 60. If timeout errors continue to be logged, you can increase this value in increments of 20 seconds.

MSCP_CMD_TMO is a DYNAMIC parameter.

3.16.7 Reclamation Parameters

On Alpha systems, nonpaged pool reclamation algorithms have been changed to ensure that packets put onto lookaside lists are returned to variable pool in a timely fashion. Both gentle and aggressive reclamation algorithms have been changed to return more packets to variable pool.

The following new parameters are related to the changed reclamation algorithms, which are explained in the *OpenVMS Version 7.1 Release Notes*.

3.16.7.1 NPAG_AGGRESSIVE Parameter (Alpha Only)

NPAG_AGGRESSIVE specifies the percentage of packets that are to remain on a nonpaged pool lookaside list after an aggressive reclamation pass. (Default: 50)

NPAG_AGGRESSIVE is a DYNAMIC parameter.

3.16.7.2 NPAG_GENTLE Parameter (Alpha Only)

NPAG_GENTLE specifies the percentage of packets that are to remain on a nonpaged pool lookaside list after a gentle reclamation pass. (Default: 85)

NPAG_GENTLE is a DYNAMIC parameter.

3.16.7.3 NPAG_INTERVAL Parameter (Alpha Only)

NPAG_INTERVAL specifies the number of seconds between gentle reclamation passes. (Default: 30)

NPAG_INTERVAL is a DYNAMIC parameter.

3.16.8 New RMS Parameters

This section presents information about the RMS_HEURISTIC and RMS_DFLRL parameters.

Note

The purpose of the RMS heuristic feature is to provide improved interoperability for existing applications using RMS to access files created by PATHWORKS. However, this is a change in behavior that may have an adverse effect on some applications.

See the *Guide to OpenVMS File Applications* and the DCL command SET RMS_DEFAULT for more information about the heuristic feature.

3.16.8.1 RMS_HEURISTIC Parameter

RMS_HEURISTIC controls whether RMS analyzes selected sequential files to determine if they are normal text or binary files. If RMS_HEURISTIC is enabled (with a value of 1), RMS analyzes sequential files with both a record format (RFM) of STM and a longest-record-length (LRL) of 0 when the files are initially opened.

By default in OpenVMS Version 7.1, the heuristic is disabled systemwide.

RMS_HEURISTIC is a DYNAMIC parameter.

3.16.8.2 RMS_DFLRL Parameter

If RMS_HEURISTIC is enabled (with a value of 1), RMS_DFLRL specifies the longest-record-length (LRL) file attribute value that RMS should use when it determines that a file with a record format of STM, STM_LF, or STM_CR and an LRL of 0 is a normal text file (rather than a binary file). See also the system parameter RMS_HEURISTIC.

The default value of RMS_DFLRL is 0; this value indicates that the internal RMS default value of 32767 is used.

RMS_DFLRL is a DYNAMIC parameter.

4
Programming Features

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

4.1 Very Large Memory Management Features (Alpha Only)

OpenVMS Alpha Version 7.1 provides extended, additional memory management features that facilitate Very Large Memory (VLM) support on OpenVMS systems. Memory-resident global sections and shared page tables provide support for database, data warehouse, and other very large database (VLDB) products. With the new extended additional VLM features, database products and data warehousing applications will be able to realize increased capacity and performance gains.

Memory-resident global sections allow a database server to keep larger amounts of ''hot'' data cached in physical memory. The database server then accesses the data directly from physical memory without performing I/O read operations from the database files on disk. With faster access to the data in physical memory, run-time performance increases dramatically.

Shared page tables allow the same database server to reduce the amount of physical memory consumed within the system. Because multiple server processes share the same physical page tables that map the large database cache, an OpenVMS Alpha system can support more server processes. This increases overall system capacity and decreases response time to client requests.

Also, with shared page tables, the database server startup time is dramatically reduced because server processes can map memory-resident global sections hundreds of times faster than traditional global sections. With a multiple gigabyte global database cache, the server startup performance gains can be significant.

For more information about using the new OpenVMS Alpha VLM features, see *OpenVMS Alpha Guide to 64-Bit Addressing and VLM Features*.

4.2 Backup Utility: New Application Programming Interface (API)

The BACKUP application programming interface (API) gives application programs access to BACKUP functions that are available to an interactive user via the DCL command BACKUP. The use of the BACKUP API provides the following benefits:

- Allows development of applications to provide more efficient management and control of archived data
- Allows development of customized BACKUP applications
- Allows development of Media Management Extension (MME) applications

Through the BACKUP API, application programs can save individual files or the contents of entire disk volume sets. The BACKUP API also enables application programs to retrieve information about files or disk and tape volumes.

An application program calls routine BACKUP$START with an argument that points to a variable-length array, which consists of option structures that specify the required BACKUP operation. Each relevant BACKUP qualifier is represented by an option structure or combination of option structures. The call to BACKUP$START in combination with the option structures in the variable-length array form the equivalent of a BACKUP command at the DCL level.

The option structure types are defined in the language definition file (for example, BAPIDEF.H).

For more information about the Backup utility, see the *OpenVMS Utility Routines Manual*.

4.3 Byte and Word Instruction Emulator: Now Part of the Installation Kit (Alpha Only)

The Byte and Word Instruction Emulator, which required a separate installation in OpenVMS Alpha Version 7.0, is now completely integrated into the OpenVMS Alpha Version 7.1 kit. This emulator executes the following Alpha processor instructions when they are not implemented directly by the microprocessor:

- LDBU and LDWU
- STB and STW
- SEXTB and SEXTW

Some new versions of Alpha compilers can generate these instructions if the code developer specifically requests it. If you run an image with these instructions in it, and your Alpha hardware does not support them, the Byte and Word Instruction Emulator will perform the specified instructions.

As a supplement to the emulator, an Emulator utility (EMULATOR_UTIL.C) is also included and is located in SYS$EXAMPLES. Comments in this file show you how to build and use the Emulator utility to perform the following tasks:

- Print the number of emulated instructions in the last image and the locations of the last 16 emulated instructions.
- Change the number of times that the SYSTEM-I-EMULATED message is printed (the default is five).
- Set the emulator back to its default state.

4.4 C Function Prototypes: Documented in the System Services Routines (Alpha Only)

With OpenVMS Alpha Version 7.0, SYS$LIBRARY:SYS$STARLET_C.TLB provided C function prototypes for system services. For OpenVMS Alpha Version 7.1, these prototypes are documented in the *OpenVMS System Services Reference Manual* along with the macro formats.

For each prototype, the manual provides the correct syntax (which shows the arguments the function accepts in the order in which it expects them), a description of each argument, and the type of data returned by the function.

4.5 DECwindows X11 Display Server: Upgraded to Version 11, Release 6

OpenVMS Version 7.1 supports the latest version of the DECwindows X11 display server which has been upgraded from Version 11, Release 5 (X11 R5) to Version 11, Release 6 (X11 R6). Included in this upgrade are the following new extensions supported by the server. Note that the version of the X Window System compatible X programming library (Xlib) currently shipped with DECwindows Motif Version 1.2-4 for OpenVMS does not support these new extensions.

- XTest Extension

  The XTest extension is actually a set of client and server extensions required for ''hands off'' validation of the X11 server. This set of extensions provides:

  – Access to ''write-only'' server resources (for example, the cursor associated with a specific window)

  – Remote activation of certain input device events (such as controlling the pointing device or pressing a key or button)

- XC-MISC Extension

  The XC-MISC extension enables clients to periodically poll the server and retrieve previously used ID ranges. This operation, performed transparently by Xlib, provides a renewable source of IDs by recycling ID ranges that are currently idle. The XC-MISC extension is especially useful for systems running applications, such as the DECwindows Motif window manager, that can often reach the upper limit of available IDs.

- BIG-REQUESTS Extension

  The BIG-REQUESTS protocol extension allows clients to extend the length field in protocol requests to a 32-bit value from the standard X protocol of 2^{18}bytes long. This is useful for the PEX 3D graphics extension and other extensions that transmit complex information to the server. Note that the current version of PEX shipped with Digital Open3D Version 3.2 does not take advantage of this extension.

- X Logical Font Description

  The X Logical Font Description now includes general 2D linear transformations, character set subsets, and support for polymorphic fonts.

4.6 Ethernet Media Types: Now Selected from the Console (Alpha Only)

Prior to OpenVMS Version 7.1, the SYS$EWDRIVER LAN device driver sensed the Ethernet media connections BNC, AUI, and Twisted Pair automatically. The LAN adapters supported by this driver include the DE435, DE450, DE500, and Tulip integral Ethernet device. Their device name under OpenVMS is EWx0, where *x* is the controller letter.

In OpenVMS Version 7.1, this driver uses a console environment variable to select the proper media connection. For each EW device recognized by the console, there is a console environment variable called EWx0_MODE, where *x* is the controller letter. This variable is set with the following command:

```
>>> SET EWx0_MODE media_selection
```

where *media_selection* is one of the following:

Twisted-Pair
Full Duplex, Twisted-Pair
AUI
BNC
Fast
FastFD (Full Duplex)
Auto-Negotiate

During the OpenVMS boot sequence, a message is sent to the operator's console that shows which media was set by the console and handed to the device driver. For example, if the EWA0_MODE console environment variable is set to FAST, the following message is displayed at the console:

```
%EWA0, Fast(100baseT) mode set by console
```

If a console environment variable is set to an unsupported media type, the driver attempts to sense or negotiate the media type automatically. This type of media sensing and negotiation, called **auto negotiation**, is supported by the DE500-AA Ethernet adapter.

In some cases, an Alpha system console might assign a controller letter to an adapter differently from OpenVMS Alpha. You should issue a SHOW CONFIGURATION command at the console to determine the correct letter designation for each adapter. In the case of different controller letter assignments, the letter designation in the message broadcast to the console by the driver may not agree with the console setting.

If the system is already running OpenVMS Alpha, you can issue the following LANCP command to select the proper media type:

```
$ MCR LANCP SET DEV EWx0/MEDIA=media_selection
```

where *media_selection* is one of the following:

AUTOSENSE
TWISTEDPAIR
AUI
BNC

4.7 Fast Ethernet Adapters: Support for the DE500-AA Adapter (Alpha Only)

Support for the DE500-AA PCI Fast Ethernet adapter has been added to OpenVMS Alpha Version 7.1. This device supports twisted pair media at speeds of 10Mb/s and 100Mb/s at full or half duplex performance. This device also supports automatic negotiation (**auto negotiation**) as proposed in the IEEE 802.3 standard.

Auto negotiation provides a method to detect the operational characteristics supported by a host device at the other end of the network link, negotiate common abilities, and configure the link accordingly. The driver can also detect changes in the link dynamically and renegotiate communication characteristics accordingly. Auto negotiation always selects the highest common operating mode possible between the two link partners. For the DE500-AA adapter, the highest operating mode is 100Mb/s full duplex unless programmed to a lesser mode.

Auto negotiation is enabled or disabled in two ways: by setting a console environment variable or by using the LANCP utility under OpenVMS. When both link partners enable auto negotiation, the OpenVMS driver selects a common operating mode, if possible. The OpenVMS driver can dynamically detect changes in the link and renegotiate a new operating mode automatically. The driver then broadcasts a message to the operator console similar to the following:

```
%EWB0, Auto Negotiation detected link down
%EWB0, Fast(100baseT) Ethernet connection selected
```

Note

For the OpenVMS driver to properly select an operating mode, both the local and remote hosts must have auto negotiation enabled.

4.7.1 Enabling Auto Negotiation with the Console Environment Variable

To enable auto negotiation using a console environment variable, issue the following command at the console:

```
>>> SET EWx0_MODE AUTO-NEGOTIATE
```

where *x* is the controller letter designation assigned to the DE500-AA adapter by the console.

To disable auto negotiation using the console environment variable, the operating mode has to be explicitly selected. The following example shows how to disable auto negotiation and enable the DE500-AA adapter to operate at 100Mb/s full duplex:

```
>>> SET EWx0_MODE FASTFD
```

where *x* is the controller letter designation assigned to the DE500-AA adapter by the console.

4.7.2 Enabling Auto Negotiation with LANCP

To enable auto negotiation under OpenVMS, issue the following LANCP command:

```
$ MCR LANCP SET DEV EWx/SPEED=AUTONEGOTIATE
```

where *x* is the controller letter designation assigned to the DE500-AA adapter by OpenVMS.

To disable auto negotiation under OpenVMS, the operating mode has to be explicitly selected. The following example shows how to disable auto negotiation and enable the DE500-AA to operate at 100Mb/s full duplex using the LANCP utility:

```
$ MCR LANCP SET DEV EWx/SPEED=100/FULLDUPLEX
```

where *x* is the controller letter designation assigned to the DE500-AA adapter by OpenVMS.

4.8 Kernel Threads Features

The following sections describe the new kernel threads features included in OpenVMS Version 7.1.

Kernel threads permit concurrent processing over all central processing units (CPUs) in a multiprocessor system by allowing a multithreaded application to have a thread executing on every CPU. Concurrent processing is comparatively faster than a single thread of execution through one CPU.

However, not all images are safe for use with threads. An application is thread-safe if threads do not interfere with each other during their independent execution. One example of interference is when two or more threads modify a common memory location without synchronization. This situation can lead to incorrect and inconsistent results. Using kernel threading, where the threads run concurrently on multiple CPUs, can greatly aggravate this problem. By disabling multiple kernel threading for a process, an application runs much less risk of encountering this problem.

Use the new kernel threads features when you have determined that your application is threads-safe and want to enable the kernel threading environment. Alternatively, you can use the THREADCP tool if the image has already been built. To find out whether an image is thread-safe, consult with the image provider.

4.8.1 Linker Utility: New /THREADS_ENABLE Qualifier

The new linker qualifier /THREADS_ENABLE allows you to turn the new kernel threads features on and off on a per-image basis.

The principal benefit of threading is to allow you to launch multiple paths of execution within a process. With threading, you can have some threads running while others are waiting for an external event to occur, such as I/O. The multi-threading kernel of OpenVMS Alpha can place threads on separate CPUs for concurrent execution; this can enable a process to run faster.

When you specify the /THREADS_ENABLE qualifier, the OpenVMS linker sets the appropriate bits in the image header of the image being linked. These bits control the following:

- Whether the image will utilize a multiple kernel threads environment
- Whether the image can receive upcalls from the OpenVMS scheduler

Note that in OpenVMS Version 7.1, only OpenVMS Alpha systems honor these control bits.

Following are explanations of how to disable the kernel threading environment for a process during the execution of images that might not be thread-safe.

Format

LINK /NOTHREADS_ENABLE (default)

LINK /THREADS_ENABLE[=(MULTIPLE_KERNEL_THREADS,UPCALLS)]

Qualifier Values

MULTIPLE_KERNEL_THREADS

This keyword sets the MULTIPLE_KERNEL_THREADS bit in the Alpha image header of the image being built. On systems that support multiple kernel threads, this bit indicates to the image activator that the image can be run in a multiple kernel threads environment.

UPCALLS

This keyword sets the UPCALLS bit in the OpenVMS image header of the image being built. On systems that support upcalls, this bit indicates to the image activator that the process can receive upcalls from the OpenVMS scheduler.

Description

The new image header bits allow you to control your threads environment on a per-process basis rather than systemwide. The image activator examines the new bits in the image header to determine the environment in which the image is to run.

Caution

The OpenVMS linker does not analyze whether the image can be safely placed in a multiple kernel threads environment. The linker sets only the control bits.

Examples

1.
```
$ LINK /NOTHREADS_ENABLE
```

   This command keeps the MULTIPLE_KERNEL_THREADS and UPCALLS bits clear in the image header. It is the default.

2.
```
$ LINK /THREADS_ENABLE
```

   This command sets the MULTIPLE_KERNEL_THREADS and UPCALLS bits in the image header of the image that is being built.

3.
```
$ LINK /THREADS_ENABLE=UPCALLS
```

   This command sets only the UPCALLS bit in the image header of the image that is being built.

4.
```
$ LINK /THREADS_ENABLE=MULTIPLE_KERNEL_THREADS
```

   This command sets only the MULTIPLE_KERNEL_THREADS bit in the image header of the image that is being built.

4.8.2 THREADCP Tool (Alpha Only)

THREADCP can be used to enable the new threads features for an image which, for whatever reason, will not be rebuilt.

The tool provides the ability to ENABLE, DISABLE, and SHOW the state of the thread control bits in an image's header.

The THREADCP command verb is not part of the normal set of DCL commands. To use the tool, you must define the command verb before invoking it.

When using this tool, an input file name is a required parameter with all qualifiers. All qualifiers and parameters can be abbreviated to the first character. When the SHOW qualifier is used alone with the THREADCP command, the file name can contain wildcard characters.

Once the THREADCP command verb is defined, the control bits can be set or cleared using the /ENABLE and /DISABLE qualifiers, respectively.

The ENABLE and DISABLE qualifiers require the names of the control bits to be modified. One or both control bits can be supplied. The user must have write access to the file. If the image is currently being executed or is installed it cannot be modified. If no control bit is specified, the default is to operate on both bits.

Examples

1.
```
$ SET COMMAND SYS$UPDATE:THREADCP.CLD
```

   This command defines the THREADCP command verb.

2.
```
$ THREADCP/SHOW TEST.EXE
```

   This command checks the current setting of the control bits for the image TEST.EXE.

3.
```
$ THREADCP/SHOW SYS$SYSTEM:*
```

   This command checks the current setting of the control bits for all SYS$SYSTEM images.

4.
```
$ THREADCP/ENABLE=(MULTIPLE_KERNEL_THREADS, UPCALLS) TEST.EXE
```

   This command sets both bits for the image TEST.EXE.

5.
```
$ THREADCP/DISABLE=(MULTIPLE_KERNEL_THREADS, UPCALLS) TEST.EXE
```

   This command clears both bits for the image TEST.EXE.

Note: In general, it is recommended that only the link methods shown in examples 1 or 2 be used.

4.9 LAN Emulation: Over an ATM Network (Alpha Only)

OpenVMS Version 7.1 supports local area network (LAN) emulation for Ethernet over asynchronous transfer mode (ATM) networks. By implementing an emulated LAN over ATM, you enable a group of ATM stations to act like a traditional LAN. You can then bring some of the benefits of ATM's speed to your Ethernet networks. Emulated LAN over ATM allows you to run your existing applications unchanged, while the computers on which your applications are running are connected to an ATM network.

ATM is a switched, connection-oriented mode of transfer, unlike Ethernet which is a shared, connectionless protocol medium.

The ATM protocol communicates by first establishing between two computers, or end points, a virtual circuit (VC) through one or more ATM switches. ATM then provides a physical path for data flow between the end points by either a permanent virtual circuit (PVC) or a switched virtual circuit (SVC).

Permanent Virtual Circuits (PVCs)

Permanent virtual circuits are established and removed by prior arrangement. They are established manually by an operator before sending any data between end points on a network. Some PVCs are defined directly on the switch; others are predefined for use in managing switched virtual circuits (SVCs).

Switched Virtual Circuits (SVC)

Switched virtual circuits require no operator interaction to create and manage connections between end points. Software dynamically establishes and removes connections as they are needed through the request of an end point.

4.9.1 Components

LAN emulation over an ATM network permits a group of ATM stations to perform as though they were connected to an ordinary local area network. With LAN emulation, existing applications can run unchanged on computers directly connected to the ATM network. The LAN emulation hides the underlying ATM network at the media access control (MAC) layer, which provides device driver interfaces.

Table 4–1 shows the four components that make up a LAN emulation over an ATM network. OpenVMS Version 7.1 supports only the LAN emulation client (LEC). Digital's GIGAswitch/ATM provides support for the LAN emulation server (LES), broadcast and unknown server (BUS), and LAN emulation configuration server (LECS).

Table 4–1 Components of LAN Emulation Over an ATM Network

Component	Function
LAN emulation client (LEC)	Provides a software driver that runs on a network client and enables LAN client applications to connect to an ATM network.
LAN emulation server (LES)	Maintains a mapping between LAN and ATM addresses by resolving LAN media access control (MAC) addresses with ATM addresses.
Broadcast and unknown server (BUS)	Maintains sessions with every LAN emulation client (LEC) in the network. For broadcast messages, the BUS sends messages to every attached LEC. The LECs then forward the message to their respectively attached LANs. For multicast messages, the BUS sends messages to only those LECs that have devices in the multicast group. For an LEC that wants to send a regular message whose destination MAC address is unknown, the BUS can be used to determine this address.
LAN emulation configuration server (LECS)	Provides a service for LAN emulation clients by helping to determine which emulated LAN each of the LEC's registered users should join, since each client can specify which emulated LAN to join.

The LEC exists on all ATM-attached computers that participate in the LAN emulation configuration. LEC provides the ATM MAC-layer connectionless function that is transparent to the LAN-type applications. The LEC, LES, and BUS can exist on one ATM-attached computer or on separate computers. Alternatively, the server functions could reside inside an ATM switch, and in fact, do exist in the GIGAswitch/ATM.

4.9.2 ATM Adapters

Table 4–2 shows the adapters that OpenVMS Version 7.1 supports for LAN emulation over ATM networks.

Table 4–2 ATM Adapters

Adapter	System Type
ATMWORKS 350	All PCI-based systems with the exception of the AlphaStation 200 and 400 series
ATMWORKS 750	All TURBOchannel-based systems with the exception of the DEC 3000-300

For more detailed information, see Section 4.9.4 and the *OpenVMS I/O User's Reference Manual*.

For information about using LANCP and system manager commands with new qualifiers for LAN emulation over ATM networks, see *OpenVMS System Management Utilities Reference Manual: A–L*, and *OpenVMS System Manager's Manual*.

4.9.3 Topography

Figure 4–1 shows the topography of a typical ATM emulated LAN.

4.9.4 Supporting LAN Emulation Over ATM

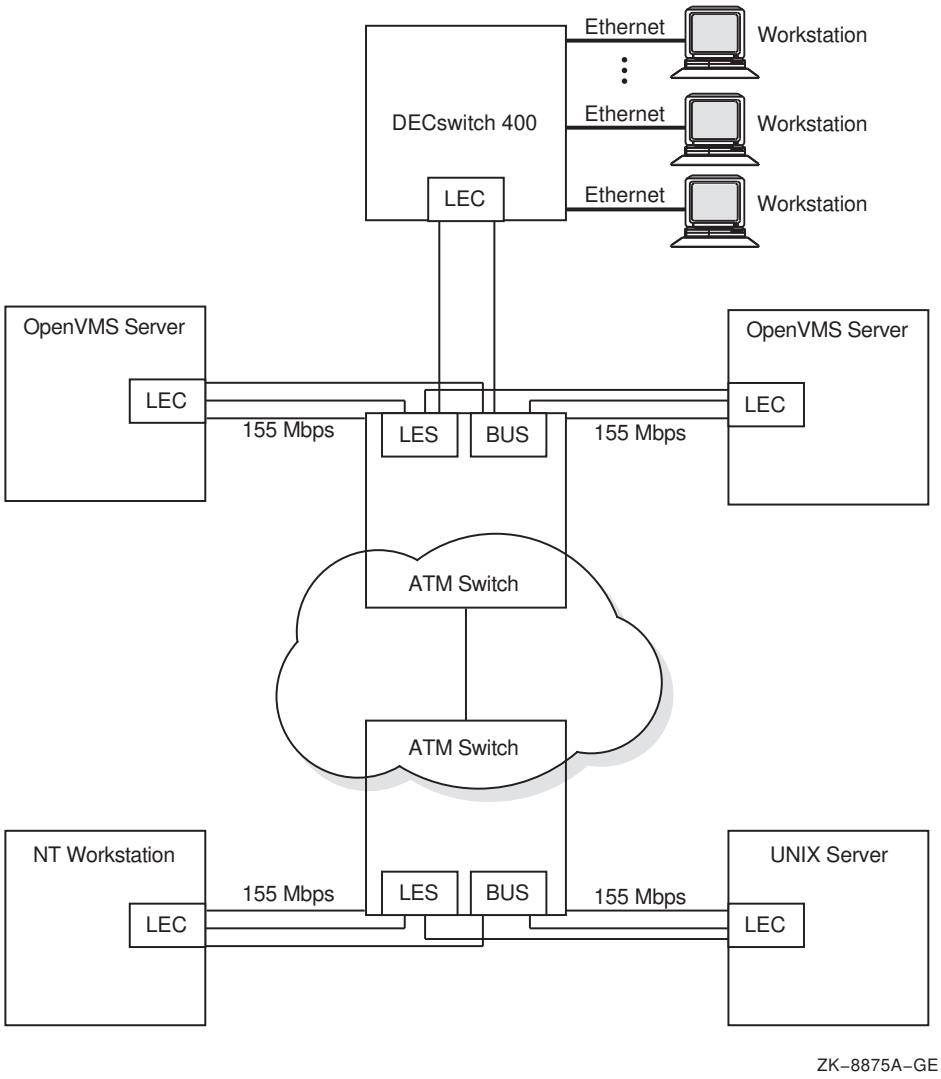
OpenVMS Version 7.1 provides LAN Emulation Client (LEC) support over the two new ATM LAN devices: ATMWORKS 350 and ATMWORKS 750. The LAN Emulation Client software supports a single IEEE/802.3 Emulated LAN per adapter, and UNI 3.0 or UNI 3.1. It also supports the maximum frame sizes of 1516, 4544, and 9234 bytes.

The ATMWORKS 350 is a 155 megabits per second (Mb/s) ATM device for PCI-bus Alpha systems with the exception of the AlphaStation 200 series, the AlphaStation 400 and the AlphaServer 400. SYS$HWDRIVER.EXE provides support for this adapter.

The ATMWORKS 750 is a 155 Mb/s ATM device for TURBOchannel Alpha systems with the exception of the DEC 3000-300. SYS$HCDRIVER.EXE provides support for this adapter.

SYS$ELDRIVER.EXE provides the Emulated LAN support, and it provides the means for communicating over the LAN ATM. The QIO and VCI interface to the Emulated LAN is the same as that described for the DESVA in the *OpenVMS I/O User's Reference Manual*, except that the device type for ELDRIVER is DT$_EL_ELAN.

Figure 4–1 Emulated LAN Topography



The device name for the Emulated LAN is:

```
ELcu
```

where *c* is the controller and *u* is the unit number (for example, ELA0).

4.9.5 Enabling SONET/SDH

The ATM drivers have the capability of operating with either SONET or SDH framing. Setting the new SYSGEN parameter, LAN_FLAGS, to 1 enables SDH framing. Setting the SYSGEN parameter, LAN_FLAGS, to 0 enables SONET framing (default). For this to take effect, the SYSGEN parameter must be specified correctly before the ATM adapter driver (HCDRIVER/HWDRIVER) is loaded.

4.9.6 Booting MOP

OpenVMS Version 7.1 does not support the ATM adapters ATMWORKS 350 and ATMWORKS 750 as boot devices. LANCP/LANACP and DECnet software does support MOP booting over the emulated LAN devices.

4.9.7 Configuring an Emulated LAN (ELAN)

The LANCP program is used to set up and verify an ELAN. If the commands are defined in the permanent database, these settings will take effect at boot time. The LANCP SET commands can be issued to start up an ELAN after the system is booted.

The following example shows the commands required for setting up an ELAN with the desired parameters. Note that some of the commands generate a console message.

```
ALPHA1> mcr lancp
LANCP> set dev ela0/elan=create

%%%%%%%%%%  OPCOM  26-MAR-1996 16:57:12.89  %%%%%%%%%%
Message from user SYSTEM on ALPHA1
LANACP LAN Services
Found LAN device ELA0, hardware address 00-00-00-00-00-00

LANCP> set dev ela0/elan=(parent=hwa0,type=csmacd,size=1516)
LANCP> set dev ela0/elan=(descr="An ATM ELAN")
LANCP> set dev ela0/elan=enable=startup

%ELDRIVER, LAN Emulation event at 26-MAR-1996 16:57:28.78
%ELDRIVER, LAN Emulation startup: Emulated LAN 1 on device ELA0

LANCP> sho dev ela/char

   Device Characteristics ELA0:
                  Value  Characteristic
                  -----  --------------
                 Normal  Controller mode
               External  Internal loopback mode
                CSMA/CD  Communication medium
                     16  Minimum receive buffers
                     32  Maximum receive buffers
                     No  Full duplex enable
                     No  Full duplex operational
            Unspecified  Line media
                     10  Line speed (megabits/second)
                 CSMA/CD  Communication medium
                  "HWA0"  Parent ATM Device
          "An ATM ELAN"  Emulated LAN Description
   39999900000000008002B  LAN Emulation Server ATM Address
   A57E80AA000302FF1300
                  Enabled  Emulated LAN State
LANCP> exit
ALPHA1>
```

4.9.8 Configuring Digital ATM Hardware

The following is a specific example of setting up the Digital GIGAswitch/ATM for emulated LAN (ELAN). The OpenVMS LAN Emulation Client (LEC) software adheres to the ATM Forum's LANE Version 1.0 specification. Hence, it works with any ATM switch that supports the ELAN server software of the LAN emulation server (LES), broadcast and unknown server (BUS), and LAN emulation configuration server (LECS).

4.9.8.1 Configuring a GIGAswitch/ATM for ELAN

The GIGAswitch firmware requires Version 1.4.3, and the GIGAswitch must also have its ELAN services enabled before the OpenVMS host can join an ELAN. A LECS, a BUS, and a LES must all be started on the GIGAswitch/ATM. By using the console of the GIGAswitch/ATM, you can log in and verify if the services are enabled, which they are in the following example:

```
GIGAswitch/ATM login: user
Password:
GIGAswitch/ATM-> bus
               Broadcast and Unknown Server Summary
     BUS Number          Status               Description
                1          enabled
GIGAswitch/ATM-> les
               LAN Emulation Server Summary
     LES Number          Status               Description
                1          enabled
GIGAswitch/ATM-> elan
                         LECS Emulated LAN Summary
  Number   Status    Name                      Default   Type   Size
       1   enabled                                *      802.3  1516
GIGAswitch/ATM->
```

If the services were not enabled, start by enabling the BUS. The following commands create BUS 1, set the frame size and the ELAN name, and enable BUS 1.

```
GIGAswitch/ATM-> bus 1
GIGAswitch/ATM-> bus 1 +f 1516
GIGAswitch/ATM-> bus 1 -e
```

This following set of commands creates LES 1, sets the frame size and the ELAN name, assigns BUS 1 to LES 1, and enables BUS 1.

```
GIGAswitch/ATM-> les 1
GIGAswitch/ATM-> les 1 +f 1516
GIGAswitch/ATM-> les 1 +bus bus1
GIGAswitch/ATM-> les 1 -e
```

Finally, the LECS is given knowledge of LES 1, the ELAN frame size, the ELAN name, and ELAN 1 is enabled.

```
GIGAswitch/ATM-> elan 1 +f 1516
GIGAswitch/ATM-> elan 1 +les les1

GIGAswitch/ATM-> elan 1 +u 10   ! max unknown frame count
GIGAswitch/ATM-> elan 1 +tu 60  ! max unknown frame time
GIGAswitch/ATM-> elan 1 +tf 4   ! flush time-out

GIGAswitch/ATM-> elan 1 -e
```

The following commands must be executed each time the GIGAswitch/ATM is rebooted unless you store the configuration in NVRAM.

```
GIGAswitch/ATM-> bus 1 +nv
GIGAswitch/ATM-> les 1 +nv
GIGAswitch/ATM-> elan 1 +nv
```

4.9.9 Configuring Digital TCP/IP Services for OpenVMS Over an Emulated LAN

The Emulated LAN is a new network device that is not recognized by the current version of Digital TCP/IP Services for OpenVMS. For this software to communicate over the Emulated LAN, the device EL must be defined by the following command:

```
UCX> DEFINE COMMUNICATION_CONTROLLER ELc0/
              TYPE=(ETHERNET)/INTERNET_INTERFACE=L
```

where *c* is the controller letter (for example, ELA0).

For more information, see the *Digital TCP/IP Services for OpenVMS Management* manual.

4.9.10 Configuring PATHWORKS Over an Emulated LAN

To use PATHWORKS with the Emulated LAN on OpenVMS Version 7.1, a logical must be defined before invoking PATHWORKS. This logical must be defined once per system boot before any invocation of PATHWORKS. The following DCL command can be used to define the required logical:

```
$ DEFINE/SYSTEM NETBIOS$DEVICE ELc0
```

where *c* is the controller letter (for example, ELA0).

4.10 MACRO–32 Compiler Built-Ins (Alpha Only)

New built-ins that allow the inclusion of the newly-designed byte and word instructions are included with this release of the MACRO–32 Compiler for OpenVMS Alpha. This section describes these new built-ins and a previously undocumented built-in, EVAX_SEXTL.

These byte and word instructions are implemented in some of the newest Alpha systems. They are also emulated in OpenVMS software.

The new built-ins and EVAX_SEXTL are shown in Table 4–3.

Table 4–3 New Alpha Instruction Built-Ins

Built-In	Operands	Description
EVAX_LDBU	<WQ,AB>	Load zero-extended byte from memory to register
EVAX_LDWU	<WQ,AQ>	Load zero-extended word from memory to register
EVAX_STB	<RQ,AB>	Store byte from register to memory
EVAX_STW	<RQ,AW>	Store word from register to memory
EVAX_SEXTB	<RQ,WB>	Sign extend byte
EVAX_SEXTW	<RQ,WW>	Sign extend word

Key to Operands

AB = address of byte
AQ = address of quadword
AW = address of word
RQ = read quadword
WB = write byte
WW = write word
WL = write longword
WQ = write quadword

(continued on next page)

Table 4–3 (Cont.) New Alpha Instruction Built-Ins

Built-In	Operands	Description
EVAX_SEXTL	<RQ,WL>	Sign extend longword

Key to Operands

AB = address of byte
AQ = address of quadword
AW = address of word
RQ = read quadword
WB = write byte
WW = write word
WL = write longword
WQ = write quadword

Note

Memory references in the MACRO-32 compiler built-ins are always assumed to be quadword aligned except in EVAX_SEXTB, EVAX_SEXTW, EVAX_LDBU, EVAX_LDWU, EVAX_STB, EVAX_STW, EVAX_LDQU, and EVAX_STQU.

The best environment in which to run code that contains the byte and word built-ins is on an Alpha computer that implements these instructions in hardware. If you run such code on an OpenVMS Alpha system that implements them by software emulation, the following limitations exist:

- Performance reduction

  The overhead of handling the exception to trigger the software emulation causes a performance loss. If software emulation is in effect, you will see the following message:

  ```
  %SYSTEM-I-EMULATED, an instruction not implemented on this processor
  was emulated
  ```

- Some capabilities not present in the software emulation

  Code that executes in inner access modes and at elevated IPL can use byte and word instructions via the emulator (in other words, activation of the software emulator above IPL2 will not cause a bugcheck).

  However, hardware limitations can result in situations where it is not always possible to use the emulator in inner-mode code. For example, direct writes to hardware control registers using byte and word instructions are impossible because they require the presence of address lines whose function cannot be emulated.

Furthermore, if the code with these built-ins executes on a system without either the byte and word software emulator or a processor that implements the byte and word instructions in hardware, it will incur a fatal exception, such as the following:

```
%SYSTEM-F-OPCDEC, opcode reserved to Digital fault at
PC=0000000000020068,PS=0000001B
```

For more information about the MACRO–32 compiler, see the *Porting VAX MACRO Code to OpenVMS Alpha*.

4.11 OpenVMS Debugger Features

This section describes the following new OpenVMS Debugger features:

- Enhanced support for byte and word emulation (Alpha only)
- Optimized DFT processing
- DUMP command
- Enhanced CALL command support for C and C++
- Enhanced support for exception handlers (SET BREAK/HANDLER)
- New output message control in screen mode (SET TERMINAL/WRAP)
- Enhancements to the DECwindows Motif Interface
- Debugger Symbol File (.DSF) file support (Alpha only)
- Enhanced Fortran-90 support
- Support for typed pointer type variables in Fortran
- New predefined register displays in screen mode

For more information, see the *OpenVMS Debugger Manual*.

4.11.1 Debug Fixup Table (DFT) Processing: Optimized to Improve Debugger Startup Time

To improve startup time, the debugger no longer processes the Debug Fixup Table (DFT) when first invoked. The debugger now processes the DFT when a module is ‘‘set.’’

4.11.2 Debugger Commands: New and Changed

The following sections describe the new and modified OpenVMS Debugger commands for OpenVMS Version 7.1.

4.11.2.1 Byte and Word Emulation: Enhanced Support (Alpha Only)

The SET BREAK command has a new /SYSEMULATE qualifier that allows you to set a breakpoint to cause the debugger to suspend program execution whenever an instruction in a specified class is emulated. An optional mask allows you to specify the class of emulated instructions that triggers the breakpoint. The only class of emulated instructions currently defined consists of the BYTE and WORD instructions, defined in the mask by bit 0.

4.11.2.2 DUMP Command

The OpenVMS Debugger has a new DUMP command to display the contents of memory in a format similar to the OpenVMS DUMP command. The new DUMP command displays the contents of memory, including registers, variables, and arrays. It accepts the following qualifiers:

- /BINARY
- /BYTE
- /DECIMAL
- /HEXADECIMAL
- /LONGWORD
- /OCTAL
- /QUADWORD
- /WORD

4.11.2.3 Enhanced CALL Command

When the current language is C or C++, the CALL command by default now passes arguments by value rather than by reference. In addition, you can now pass the following arguments without using a passing mechanism lexical (such as %REF or %VAL):

- Routine references
- Quoted strings (treated as %REF strings)
- Structures, records, and objects
- Floating-point parameters (F_Float, D_Float, G_Float, S_Float, T_Float) by value

4.11.2.4 SET BREAK/HANDLER Qualifier

The debugger SET BREAK command has a new /HANDLER qualifier that causes the debugger to scan the call stack and attempt to set a breakpoint on every established frame-based handler whenever the program being debugged has an exception. The debugger does not discriminate between standard run-time library (RTL) handlers and user-defined handlers.

On Alpha systems, many RTLs establish a jacket RTL handler on a frame where the user program has defined its own handler. This RTL jacket does some setup and argument manipulation before actually calling the handler defined by the user. When processing the exception, the debugger sets the breakpoint on the jacket RTL jacket handler, because that is the address on the call stack. If the debugger suspends program execution at a jacket RTL handler, you can usually reach the user-defined handler by entering a STEP/CALL command followed by a STEP/IN command. Some cases might require that you enter additional sequences of STEP/CALL and STEP/IN commands.

If the jacket RTL handler is part of an installed shared image such as ALPHA LIBOTS, the debugger cannot set a breakpoint on it. In this case, activate the RTL as a private image by defining the RTL as a logical name. For example:

```
DEFINE LIBOTS SYS$SHARE:LIBOTS.EXE;
```

This technique bypasses the installed image lookup procedure. Note that the semicolon that follows the .EXE file extension is required.

4.11.2.5 SET TERMINAL/WRAP Qualifier

In screen mode, the SET TERMINAL command has a new /WRAP qualifier to control where the debugger inserts a carriage return and linefeed to format an output message in the message output display. The /WRAP qualifier sets the wrap column at the current value established by the /WIDTH qualifier. For example:

```
SET TERM/WIDTH:20/WRAP
```

4.11.3 DECwindows Motif Interface: Now Has Two Wait Cursors

The DECwindows Motif interface to the debugger now has two different wait cursors that indicate wait states. One wait cursor is the standard clock cursor. This cursor indicates that the debugger is performing an action as a result of a user command. The second wait cursor is a spider. This cursor indicates that the program being debugged is executing.

4.11.4 Debugger Symbol File (.DSF) File Support (Alpha Only)

On Alpha systems, you can now debug programs that have been linked with the /NODEBUG/DSF=filespec qualifiers. You must be running in a configuration in which DBG$PROCESS is defined as DEFAULT or MULTIPROCESS, and you must define DBG$IMAGE_DSF_PATH to point to the directory where the required .DSF files reside. Note that the only .DSF files in this directory must be those required to debug the target image.

4.11.5 Fortran Debugging: Enhanced Operator and Variable Support

The debugger now supports the following Fortran operators:

Operator	Meaning
==	equal to
/=	not equal to
<	less than
>	greater than
<=	less than or equal to
>=	greater than or equal to
%	field delimiter

The debugger also supports *typed pointer type* variables in Fortran. This allows users to manipulate Fortran pointer variables.

4.11.6 Register Displays: Predefined in Screen Mode

On Alpha systems, the new predefined display REG contains, in hexadecimal format, general-purpose registers R0 to R28, FP (R29), SP (R30), R31, PC, PS, floating-point registers F0 to F31, and as many of the top call-stack values as will fit in the display.

On Alpha systems, the new predefined display IREG contains, in hexadecimal format, general-purpose registers R0 to R28, FP (R29), SP (R30), R31, PC, PS, and as many of the top call stack values as can be displayed in the window.

On Alpha systems, the new predefined display FREG contains floating-point registers F0 to F31, displayed in floating-point format, FPCR, SFPCR, and as many of the top call stack values (in hexadecimal format) as can be displayed in the window.

On VAX systems, the new predefined display IREG contains the same information as the predefined REG display, but the display window is configured differently. Display IREG is included in the VAX implementation to allow debugger .COM and .INI files to be ported more easily between VAX and Alpha debugging applications.

4.12 OpenVMS SCSI Device Drivers: Enhanced for OpenVMS Version 7.1

OpenVMS SCSI device drivers have been enhanced significantly for OpenVMS Version 7.1. These enhancements allow OpenVMS SCSI device drivers to work with a wider range of non-Digital SCSI devices, while taking advantage of as much SCSI functionality as the devices offer. In most cases, the OpenVMS Version 7.1 SCSI driver enhancements eliminate the errors that occurred when some non-Digital devices were mounted after booting Version 6.2 of the OpenVMS operating system.

4.13 SCSI Skip Files: New Modifier for SCSI IO$_SKIPFILE Function

As of OpenVMS Version 7.1, a new modifier (IO$M_ALLOWFAST) can be used with the IO$_SKIPFILE function to provide better performance on SCSI tape drives that support the SCSI space-by-file-marks command and the SCSI read position command. OpenVMS ignores the new modifier for nonSCSI tape drives.

The IO$M_ALLOWFAST modifier allows a SCSI tape subsystem to use the optimized IO$_SKIPFILE if it is capable. If a specific tape device does not adequately support the optimized IO$_SKIPFILE that uses the SCSI space-by-file-marks command, the tape subsystem will use the standard space-by-records algorithm.

The new optimized IO$_SKIPFILE mechanism requires tape drives that support the SCSI Space command, with the filemarks code, and the SCSI Read Position command. The following tape drives that do not support the new IO$_SKIPFILE mechanism include the following: TSZ05, TSZ07, TZK10, TKZ60, TKZ09, TZ30.

The new IO$M_ALLOWFAST modifier provides slightly different semantics for IO$_SKIPFILE operations (refer to *OpenVMS I/O User's Reference Manual*). The MTAACP and Backup utilities are compatible with the new behavior and have been modified to use the new IO$M_ALLOWFAST modifier. Other tape applications should be examined to determine whether they are compatible with the new semantics. If an application is compatible, or if it can be made compatible, the application should be modified to use the IO$M_ALLOWFAST modifier with the IO$_SKIPFILE function. If you have an application that is compatible with the new semantics, but it can not be modified to use the IO$M_ALLOWFAST modifier, refer to the documentation in SYS$ETC:MKSET.TXT.

For more information about how to use the IO$M_ALLOWFAST modifier with the IO$_SKIPFILE function, see the *OpenVMS I/O User's Reference Manual*.

4.14 System Dump Analyzer (SDA) Features (Alpha Only)

The following sections describe the new OpenVMS Alpha SDA commands, qualifiers, and enhancements available in this software release.

4.14.1 New Commands

The following sections describe the new OpenVMS Alpha SDA commands and their parameters and qualifiers.

4.14.1.1 SET ERASE_SCREEN Command

The SET ERASE_SCREEN command enables or disables the automatic clearing of the screen before each new page of SDA output. It has two parameters and no qualifiers. Table 4–4 shows the two parameters and their meanings.

Table 4–4 SET ERASE_SCREEN Parameters

Parameter	Meaning
off	Disables the usual clear screen action and replaces it with a blank line. With SDA's usual behavior, SDA clears the screen and then shows the data. This action does not affect what is written to a file where the SET LOG or SET OUTPUT is used. This command affects only the screen.
on	Enables the automatic clearing of the screen before each new page of SDA output.

For more detailed information, see the *OpenVMS Alpha System Dump Analyzer Utility Manual*.

4.14.1.2 SHOW ADDRESS Command

The SHOW ADDRESS command displays page table related information about a memory address. It has one parameter and one qualifier. Table 4–5 shows the one parameter and its meaning.

Table 4–5 SHOW ADDRESS Parameter

Parameter	Meaning
address	Shows where an address is mapped at different page table entry levels.

Table 4–6 shows the one qualifier and its meaning.

Table 4–6 SHOW ADDRESS Qualifier

Qualifier	Meaning
/PHYSICAL	Indicates that a physical address has been given. The SHOW ADDRESS command displays the virtual address that maps to the given physical address.

Examples

1.
```
SDA> SHOW ADDRESS 80000000

FFFFFFFF80000000 is an S0/S1 address

Mapped by Level-3 PTE at: FFFFFFFDFFE00000
Mapped by Level-2 PTE at: FFFFFFFDFF7FF800
Mapped by Level-1 PTE at: FFFFFFFDFF7FDFF0
Mapped by Selfmap PTE at: FFFFFFFDFF7FDFF0

Also mapped in SPT window at: FFFFFFFFFFDF0000
```

   The SHOW ADDRESS command in this example shows where the address 80000000 is mapped at different page table entry levels.

2.
```
SDA> SHOW ADDRESS 0

0000000000000000 is a P0 address

Mapped by Level-3 PTE at: FFFFFFFC00000000
Mapped by Level-2 PTE at: FFFFFFFDFF000000
Mapped by Level-1 PTE at: FFFFFFFDFF7FC000
Mapped by Selfmap PTE at: FFFFFFFDFF7FDFF0
```

   The SHOW ADDRESS command in this example shows where the address 0 is mapped at different page table entry levels.

3.
```
SDA> SHOW ADDRESS FFFFFFFDFF000000

FFFFFFFDFF000000 is the address of a process-private Level-2 PTE

Mapped by Level-1 PTE at: FFFFFFFDFF7FC000
Mapped by Selfmap PTE at: FFFFFFFDFF7FDFF0

Range mapped at level 2: FFFFFFFC00000000 to FFFFFFFC00001FFF (1 page)
Range mapped at level 3: 0000000000000000 to 00000000007FFFFF (1024 pages)
```

   The SHOW ADDRESS command in this example shows where the address FFFFFFFDFF7FC000 is mapped at page table entry and the range mapped by the PTE at this address.

For more detailed information, see the *OpenVMS Alpha System Dump Analyzer Utility Manual*.

4.14.1.3 SHOW BUGCHECK Command

The SHOW BUGCHECK command displays the following bugcheck codes: value, name, and text. It has two parameters and one qualifier. Table 4–7 shows the parameters and their meanings.

Table 4–7 SHOW BUGCHECK Parameters

Parameter	Meaning
name	Displays the named bugcheck code.
number	Displays the requested bugcheck code.

Table 4–8 shows the one qualifier.

Table 4–8 SHOW BUGCHECK Qualifier

Qualifier	Meaning
/ALL	Displays all bugcheck codes. It is the default.

Examples

1.
```
SDA> SHOW BUGCHECK 100
0100  DIRENTRY       ACP failed to find same directory entry
```

   The SHOW BUGCHECK command in this example shows the requested bugcheck by number.

2.
```
SDA> SHOW BUGCHECK DECNET
08D0  DECNET         DECnet detected a fatal error
```

   The SHOW BUGCHECK command in this example shows the requested bugcheck by name.

3.
```
SDA> SHOW BUGCHECK

BUGCHECK codes and texts
------------------------
0008  ACPMBFAIL      ACP failure to read mailbox
0010  ACPVAFAIL      ACP failure to return virtual address space
0018  ALCPHD         Allocate process header error
0020  ALCSMBCLR      ACP tried to allocate space already allocated
   .
   .
   .
```

   The SHOW BUGCHECK command in this example shows the requested bugcheck by displaying all codes.

For more detailed information, see the *OpenVMS Alpha System Dump Analyzer Utility Manual*.

4.14.1.4 SHOW GLOBAL_SECTION_TABLE Command

The SHOW GLOBAL_SECTION_TABLE command displays information contained in the global section table. There are no parameters and one qualifier. Table 4–9 shows the qualifier and its meaning.

Table 4–9 SHOW GLOBAL_SECTION_TABLE Qualifier

Qualifier	Meaning
/SECTION_INDEX=*n*	Displays the global section table entry for the specified section.

This new command is equivalent to the SHOW PROCESS /SYSTEM /PROCESS_ SECTION_TABLE command.

Example

```
SDA> SHOW GST

Global Section Table
--------------------

Global section table information
--------------------------------

        Last entry allocated     0187
        First free entry         0000

Global section table
--------------------
INDEX  ADDRESS    SECT/GPTE ADDR     PAGELETS   WINDOW     VBN      CCB/GSD    REFCNT   FLINK BLINK   FLAGS
0001   80D09FD8  FFFFFFFF.82A24000   00000069  80D202C0  00000003  00000000  00000007  0000  0000
0002   80D09FB0  FFFFFFFF.82FE0000   00000160  80D73B80  00000428  00000000  00000016  0000  0000
0003   80D09F88  FFFFFFFF.82A5A000   0000005F  80D206C0  0000014F  00000000  00000006  0000  0000
0004   80D09F60  FFFFFFFF.829A8000   00000001  80D73B80  0000058B  00000000  00000001  0000  0000   WRT CRF
0005   80D09F38  FFFFFFFF.82A6E000   00000009  80D21080  00000027  00000000  00000001  0000  0000
0006   80D09F10  FFFFFFFF.82998000   00000008  80D73D00  00000005  00000000  00000001  0000  0000
0007   80D09EE8  FFFFFFFF.82A76000   0000009B  80D21240  0000015A  00000000  0000000A  0000  0000
0008   80D09EC0  FFFFFFFF.829B0000   00000013  80D73EC0  00000003  00000000  00000002  0000  0000
000A   80D09E70  FFFFFFFF.8300C000   00000228  80D74080  00000002  00000000  00000015  0000  0000   WRT CRF
000B   80D09E48  FFFFFFFF.82AB0000   00000012  80D25280  000000A0  00000000  00000002  0000  0000
000C   80D09E20  FFFFFFFE.00052010   000001C2  80D88900  0000006F  81782030  00000000  000C  000C   GBL
                                    NAME = INS$81781FC0_003
000D   80D09DF8  FFFFFFFF.82ABA000   00000059  80D26880  00000043  00000000  00000006  0000  0000
000E   80D09DD0  FFFFFFFE.00052108   00000021  80D90E40  0000000E  81782EB0  00000000  000E  000E   GBL
                                    NAME = INS$81782E30_003
000F   80D09DA8  FFFFFFFF.82ACE000   00000025  80D27E40  00000022  00000000  00000003  0000  0000
0010   80D09D80  FFFFFFFE.00052130   00000058  80D90F80  0000001B  81783280  00000000  0010  0010   GBL
                                    NAME = INS$81783210_003
0011   80D09D58  FFFFFFFF.82ADA000   000001C7  80D2B100  00000046  00000000  0000001D  0000  0000
0012   80D09D30  FFFFFFFE.00052170   000000AE  80D91BC0  00000038  81783690  00000000  0012  0012   GBL
                                    NAME = INS$81783620_003
0013   80D09D08  FFFFFFFF.82B22000   00000029  80D2C6C0  00000007  00000000  00000003  0000  0000
0014   80D09CE0  FFFFFFFE.000521D8   0000002F  80D92000  0000000E  81783A80  00000000  0014  0014   GBL
                                    NAME = INS$81783A10_003
0015   80D09CB8  FFFFFFFE.00052200   00000161  80D92300  000000B4  81783EA0  00000000  0015  0015   GBL
                                    NAME = INS$81783E20_003
0016   80D09C90  FFFFFFFF.82B36000   0000005C  80D2E440  00000024  00000000  00000006  0000  0000
0017   80D09C68  FFFFFFFE.000522C8   00000170  80D92300  00000267  81783EF0  00000000  0017  0017   GBL
                                    NAME = INS$81783E20_008
0018   80D09C40  FFFFFFFF.82B46000   000000AB  80D2FA00  0000006B  00000000  0000000B  0000  0000
   .
   .
   .
```

ZK–8829A–GE

For more detailed information, see the *OpenVMS Alpha System Dump Analyzer Utility Manual*.

4.14.1.5 SHOW GSD Command

The SHOW GSD command displays information contained in the global section descriptors. It has no parameters and four qualifiers. Table 4–10 shows the qualifiers and their meanings.

Table 4–10 SHOW GSD Qualifiers

Qualifiers	Meaning
/ALL	Displays information in all global section descriptors; that is, the system, group, and deleted global section descriptors. This qualifier is the default.
/SYSTEM	Displays information in the system global section descriptors.
/GROUP	Displays information in the group global section descriptors.

(continued on next page)

Table 4–10 (Cont.) SHOW GSD Qualifiers

Qualifiers	Meaning
/DELETED	Displays information in the deleted global section descriptors; that is, a delete pending.

Example

```
SDA > SHOW GSD

System Global Section Descriptor List
-------------------------------
                                                                                  ----------PFNMAP----------
--
ADDRESS    NAME        GSTX    FLAGS                                          BASEPFN    PAGES     REFCNT

817DAF30   SECIDX_422  02DD    0082C3C9   WRT AMOD=USER PERM
817DAE60   SECIDX_421  02DC    008A83CD   DZRO WRT AMOD=USER PAGFIL
817DAD90   SECDIX_420  02DB    0088C3CD   DZRO WRT AMOD=USER PERM PAGFIL
817DACC0   SECDIX_419  02DA    008883DC   DZRO WRT AMOD=USER PAGFIL
817DABE0   SECIDX_418  0000    0001C3C1   AMOD=USER PERM                     00000B0B   00000002   00000000
817DAB00   SECIDX_417  0000    0001C3C1   AMOD=USER PERM                     00000B0B   00000002   00000000
817DA890   SECIDX_412  02D6    0080C3CD   DZRO WRT AMOD=USER PERM
817DA850   SECIDX_411  02D5    008083CD   DZRO WRT AMOD=USER
    .
    .
    .
                                                                                                  ZK-8830A-GE
```

For more detailed information, see the *OpenVMS Alpha System Dump Analyzer Utility Manual*.

4.14.1.6 SHOW RMD Command

The SHOW RMD command displays information contained in the reserved memory descriptors. Reserved memory is used within the system by memory-resident global sections. It has two qualifiers. Table 4–11 shows the qualifiers and their meanings.

Table 4–11 SHOW RMD Qualifiers

Qualifiers	Meaning
/ADDRESS=*n*	Displays a specific reserved memory descriptor entry, given its address.
/ALL	Displays information in all the reserved memory descriptors. This qualifier is the default.

Example

```
SDA> SHOW RMD
Reserved Memory Descriptor List
------------------------------

ADDRESS  NAME       GROUP   PFN     COUNT    IN_USE  ZERO_PFN  FLAGS

80D21200 MILORD2     -S-  00000000 00000100 00000000 00000000 00000000
80D21100 MILORD1     -S-  00000A00 00000080 00000000 00000A00 00000001 ALLOC
80D21280 MILORD2     -S-  00000000 00000001 00000000 00000000 00000040 PAGE_TABLES
80D21180 MILORD1     -S-  00000180 00000001 00000000 00000180 00000041 ALLOC PAGE_TABLES
```

For more detailed information, see the *OpenVMS Alpha System Dump Analyzer Utility Manual*.

4.14.1.7 SHOW WORKING_SET_LIST Command

The SHOW WORKING_SET_LIST command displays the system working set list and retains the current process context. This command has no parameters or qualifiers. It does have, however, three options available. Table 4–12 shows the options for this command.

Table 4–12 Options for the SHOW WORKING_SET_LIST Command

Options	Meaning
WORKING_SET_LIST = GPT	Displays only working set list entries that are for global page table pages.
WORKING_SET_LIST = SYSTEM	Displays only working set list entries for pageable system pages.
WORKING_SET_LIST = LOCKED	Displays only working set list entries for pageable system pages that are locked in the system working set.

This new command is equivalent to the SHOW PROCESS /SYSTEM /WORKING_ SET_LIST command.

For more detailed information, see the *OpenVMS Alpha System Dump Analyzer Utility Manual*.

4.14.1.8 VALIDATE PFN_LIST Command

The VALIDATE PFN_LIST command validates that the page counts on lists are correct. It has no parameters and five qualifiers. Table 4–13 shows the qualifiers.

Table 4–13 VALIDATE PFN_LIST Qualifiers

Qualifiers	Meaning
/ALL	Validates all the PFN lists: bad, free, modified, and zero.
/BAD	Validates the bad page list.
/FREE	Validates the free page list.
/MODIFIED	Validates the modified page list.
/ZERO	Validates the zero page list.

For more detailed information, see the *OpenVMS Alpha System Dump Analyzer Utility Manual*.

4.14.2 New Qualifiers, Options, and Parameters

The following sections present the new OpenVMS Alpha SDA qualifiers, options, and parameters.

4.14.2.1 EVALUATE Qualifier

There is one new qualifier for the EVALUATE command. Table 4–14 shows the qualifier and its meaning.

Table 4–14 EVALUATE Command New Qualifier

Qualifier	Meaning
/TIME	Interprets and displays the expression as a 64-bit time value. Positive values are interpreted as absolute time; negative values are interpreted as delta time.

For more detailed information, see the *OpenVMS Alpha System Dump Analyzer Utility Manual*.

4.14.2.2 FORMAT Qualifier

There is one new qualifier for the FORMAT command. Table 4–15 shows the qualifier and its meaning.

Table 4–15 FORMAT Command New Qualifier

Qualifier	Meaning
/PHYSICAL	Specifies that the location given is a physical address.

For more detailed information, see the *OpenVMS Alpha System Dump Analyzer Utility Manual*.

4.14.2.3 READ Qualifiers

There are two new qualifiers for the READ command. Table 4–16 shows the qualifiers and their meanings.

Table 4–16 READ Command New Qualifiers

Qualifier	Meaning
/LOG	Outputs for each symbol table file %SDA-I-READSYM. The qualifier /LOG enables the %SDA-I-READSYM capability, which is the default.
/NOLOG	Suppresses the %SDA-I-READSYM message normally output by the SDA READ command for each symbol table file.

For more detailed information, see the *OpenVMS Alpha System Dump Analyzer Utility Manual*.

4.14.2.4 SEARCH Qualifier

There is one new qualifier for the SEARCH command. Table 4–17 shows the qualifier and its meaning.

Table 4–17 SEARCH Command New Qualifier

Qualifier	Meaning
/PHYSICAL	Specifies that the addresses used to define the range of locations to be searched are physical addresses.

For more detailed information, see the *OpenVMS Alpha System Dump Analyzer Utility Manual*.

4.14.2.5 SET FETCH Qualifiers

There are two new qualifiers for the SET FETCH command. Table 4–18 shows the qualifiers and their meanings.

Table 4–18 SET FETCH Command New Qualifiers

Qualifier	Meaning
/PHYSICAL	Sets the default access method to physical addresses.
/VIRTUAL	Sets the default access method to virtual addresses.

For more detailed information, see the *OpenVMS Alpha System Dump Analyzer Utility Manual*.

4.14.2.6 SHOW LAN Qualifiers

There are four new qualifiers for the SHOW LAN command. Table 4–19 shows the qualifiers and their meanings.

Table 4–19 SHOW LAN Command New Qualifiers

Qualifier	Meaning
/ELAN	Specifies information from an Emulated LAN (ELAN) that runs over an asynchronous transfer mode (ATM) network. The /ELAN qualifier displays the LAN station block (LSB) address, device state, and the LSB fields pertinent to an ELAN for both the parent ATM device and the ELAN psuedodevice drivers. It also specifies the name, description, parent device, state, and LAN emulation client (LEC) attributes of the ELAN. The qualifier /ELAN used with the device qualifier (/LAN /device=ELA) only displays information for the specified device or psuedodevice.
/ICOUNTERS	Specifies internal counters of the drivers by displaying the internal counters. If the /ICOUNTERS qualifier is used with the /DEVICE qualifier, the /ICOUNTERS qualifier specifies the internal counters of a specific driver.
/QUEUE	Specifies a listing of all queues, whether their status is valid or invalid, and all elements of the queues. If the /QUEUE qualifier is used with the /DEVICE qualifier, the /QUEUE specifies a specific queue.

(continued on next page)

Table 4–19 (Cont.) SHOW LAN Command New Qualifiers

Qualifier	Meaning
/VCI	Specifies the VMS Communication Interface Block (VCIB) for each LAN device with an active VCI user. If the /VCI qualifier is used with the /DEVICE qualifier, the VCIB is only displayed for the specified device.

An example of the qualifier /VCI is as follows:

Example

```
SDA> SHOW LAN/VCI/DEVICE=ICB

               -- ICB VCI Information 17-APR-1996 14:22:07 --

LSB address  = 80A1D580
Device state = 00000003 RUN,INITED

         -- ICB2 80-41 (LAST) VCI Information 17-APR-1996 14:22:07 --

VCIB address =    8096F238
CLIENT flags:     00000001 RCV_DCB
LAN flags:        00000004 LAN_INIT
DLL flags:        00000005 XMT_CHAIN,PORT_STATUS
UCB status:       00000015 STRTN,UNIQ,STRTD

VCI ID                         LAST    VCI version               00010001
UCB address                80A4C5C0    DP VCRP address           00000000
Hardware address  00-00-93-08-52-CF    LDC address               80A1D720
Physical address  00-00-93-08-52-CF    LAN medium                      TR
Transmit available         80A1D670    Outstanding operations           0
Maximum receives                  0    Outstanding receives             0
Max xmt size                   4444    Header size                     52
Build header rtn           808BF230    Report event rtn          86327130
XMT initiate rtn           808BF200    Transmit complete rtn     86326D80
XMT frame rtn              808BF210    Receive complete rtn      86326A80

      -- ICB2 80-41 (LAST) VCI Information (cont) 17-APR-1996 14:22:07 --

Portmgmt initiate rtn      808BF0C0    Portmgmt complete rtn     86327100
Monitor request rtn        00000000    Monitor transmit rtn      00000000
Monitor flags              00000000    Monitor receive rtn       00000000
Port usable                00000000    Port unusable             00000000
```

The SHOW LAN/VCI/DEVICE=ICB command displays the VCIB for a Token Ring device (ICB) which has an active VCI user (LAST).

Examples of the qualifier /ELAN are as follows:

Examples

1. SDA> SHOW LAN/ELAN

```
        -- HCA Emulated LAN LSB Information 17-APR-1996 14:08:02 --

LSB address = 8098D200
Device state = 00000101 RUN,RING_AVAIL

Driver CM VC setup adr   808986A0    Driver CM VC teardown adr    80898668
NIPG CM handle adr       8096C30C    NIPG CM SVC handle           00000000
NIPG CM agent handle adr 809B364C    NIPG CM mgr lineup handle    809B394C
NIPG CM ILMI IO handle   809B378C    MIB II handle adr            809B94CC
MIB handle adr           809B3ACC    Queue header for EL LSBs     00000000
DEC MIB handle adr       809BBD8C    NIPG current TQEs used       00000000
Count of allocated TQEs  0000000D    NIPG current pool used       0000D2C0
NIPG pool allocations    00075730

        -- ELA Emulated LAN LSB Information 17-APR-1996 14:08:02 --

LSB address = 80AB08C0
Device state = 00000001 RUN

ELAN name = ELAN 1
ELAN description = ATM ELAN
ELAN parent = HCA0
ELAN state = 00000001 ACTIVE

MAX transmit size     MTU_1516          ELAN media type      LAN_802_3
LEC attr buff adr     80AB1FC0          LEC attr buff size    00000328
Event mask            00000000          PVC identifer         00000000
Extended sense        00000000

         -- ELA Emulated LAN LEC Attributes 17-APR-1996 14:08:02 --

LAN type          00000000          LAN MTU              00000001
Proxy flag        00000000          Control timeout      0000000A
Max UF count      00000001          Max UF time          00000001
VCC timeout       000004B0          Max retry count      00000002
LEC id            00000002          Forw delay time      0000000F
Flush timeout     00000004          Path switch delay    00000006
SM state          00000070          Illegal CTRL frames  00000000
CTRL xmt failures 00000000          CTRL frames sent     0000000C
CTRL frames_rcvd  00000012          LEARPs sent          00000000
LEARPS rcvd       00000000          UCASTs sent direct   00000000
UCASTs flooded    00000006          UCASTs discarded     00000001
NUCASTs sent      00000000
Local ESI         00000000.00000000
BUS ATM addr      39999900000000008002BA57E80.AA000302FF12.00
LES ATM addr      39999900000000008002BA57E80.AA000302FF14.00
My ATM addr       39999900000000008002BA57E80.08002B2240A0.00
```

The SHOW LAN/ELAN command displays information for the parent ATM device (HCA) driver and the ELAN psuedodevice (ELA) driver.

2. SDA> SHOW LAN/ELAN/DEVICE=ELA

```
        -- ELA Emulated LAN LSB Information 17-APR-1996 14:08:22 --

LSB address = 80AB08C0
Device state = 00000001 RUN

ELAN name = ELAN 1
ELAN description = ATM ELAN
ELAN parent = HCA0
ELAN state = 00000001 ACTIVE
```

```
MAX transmit size     MTU_1516          ELAN media type      LAN_802_3
LEC attr buff adr     80AB1FC0          LEC attr buff size    00000328
Event mask            00000000          PVC identifer         00000000
Extended sense        00000000

         -- ELA Emulated LAN LEC Attributes 17-APR-1996 14:08:22 --

LAN type              00000000          LAN MTU               00000001
Proxy flag            00000000          Control timeout       0000000A
Max UF count          00000001          Max UF time           00000001
VCC timeout           000004B0          Max retry count       00000002
LEC id                00000002          Forw delay time       0000000F
Flush timeout         00000004          Path switch delay     00000006
SM state              00000070          Illegal CTRL frames   00000000
CTRL xmt failures     00000000          CTRL frames sent      0000000C
CTRL frames_rcvd      00000012          LEARPs sent           00000000
LEARPS rcvd           00000000          UCASTs sent direct    00000000
UCASTs flooded        00000006          UCASTs discarded      00000001
NUCASTs sent          00000000
Local ESI             00000000.00000000
BUS ATM addr          39999900000000008002BA57E80.AA000302FF12.00
LES ATM addr          39999900000000008002BA57E80.AA000302FF14.00
My ATM addr           39999900000000008002BA57E80.08002B2240A0.00
```

The SHOW LAN/ELAN/DEVICE=ELA command displays information for the ELAN psuedodevice (ELA) driver only.

3. `SDA> SHOW LAN/ELAN/DEVICE=HCA`

```
       -- HCA Emulated LAN LSB Information 17-APR-1996 14:08:25 --

LSB address = 8098D200
Device state = 00000101 RUN,RING_AVAIL

Driver CM VC setup adr   808986A0    Driver CM VC teardown adr    80898668
NIPG CM handle adr       8096C30C    NIPG CM SVC handle           00000000
NIPG CM agent handle adr 809B364C    NIPG CM mgr lineup handle    809B394C
NIPG CM ILMI IO handle   809B378C    MIB II handle adr            809B94CC
MIB handle adr           809B3ACC    Queue header for EL LSBs     00000000
DEC MIB handle adr       809BBD8C    NIPG current TQEs used       00000000
Count of allocated TQEs  0000000D    NIPG current pool used       0000D2C0
NIPG pool allocations    000757B2
```

The SHOW LAN/ELAN/DEVICE=HCA command displays information for the ATM device (HCA) driver only.

For more detailed information, see the *OpenVMS Alpha System Dump Analyzer Utility Manual*.

4.14.2.7 SHOW PFN_DATA Qualifiers

There are three new qualifiers for the SHOW PFN_DATA command. Table 4–20 shows the qualifiers and their meaning.

Table 4–20 SHOW PFN_DATA Command New Qualifiers

Qualifier	Meaning
/BAD	Displays the bad-page list. SDA precedes the list with a count of the pages it contains, its low limit, and its high limit.
/COLOR [={*n*|ALL}]	Displays data on page coloring. See Table 4–21 for the command options available with this qualifier.

(continued on next page)

Table 4–20 (Cont.) SHOW PFN_DATA Command New Qualifiers

Qualifier	Meaning
/ZERO	Displays the contents of the zeroed free page list.

Table 4–21 shows the command options that the /COLOR qualifier supports.

Table 4–21 Command Options with the /COLOR Qualifier

Options	Meaning
/COLOR with no value	Displays a summary of the lengths of the colored page lists for both free pages and zeroed pages.
/COLOR=*n* where *n* is a color number	Displays the data in the PFN lists (for the specified color) for both free and zeroed pages.
/COLOR=ALL	Displays the data in the PFN lists (for all colors) for both free and zeroed free pages.
/COLOR=*n* or /COLOR=ALL with /FREE or /ZERO	Displays only the data in the PFN list (for the specified color or all colors) for either free or zeroed free pages as appropriate. The qualifiers /BAD and /MODIFIED are ignored with /COLOR=*n* and /COLOR=ALL.
/COLOR without an option specified together with one or more of /FREE, /ZERO, /BAD, or /MODIFIED	Displays the color summary in addition to the display of the requested list(s).

For more information on page coloring, see the *OpenVMS System Management Utilities Reference Manual: M–Z.*

For more detailed information about the SHOW PFN_DATA command, see the *OpenVMS Alpha System Dump Analyzer Utility Manual*.

4.14.2.8 SHOW POOL Qualifier

There is one new qualifier for the SHOW POOL command. Table 4–22 shows the qualifier and its meaning.

Table 4–22 SHOW POOL Command New Qualifier

Qualifier	Meaning
/BAP	Displays the contents of the bus-addressable dynamic pool currently in use.

For more detailed information, see the *OpenVMS Alpha System Dump Analyzer Utility Manual*.

4.14.2.9 SHOW PROCESS Qualifiers

There are two new qualifiers for the SHOW PROCESS /PAGE_TABLES command. Table 4–23 shows the qualifiers and their meanings.

Table 4–23 SHOW PROCESS /PAGE_TABLES Command New Qualifiers

Qualifier	Meaning
/PAGE_TABLES = ALL	Displays the entire page table for the process from address zero to the end of process-private page table space.
/PAGE_TABLES /PT	Displays the page tables of the page table region.

For more detailed information, see the *OpenVMS Alpha System Dump Analyzer Utility Manual*.

4.14.2.10 SHOW PROCESS Options

There are five new options for the SHOW PROCESS /WORKING_SET_LIST command. Table 4–24 shows the options and their meanings.

Table 4–24 SHOW PROCESS /WORKING_SET_LIST Command New Options

Option	Meaning
/WORKING_SET_LIST = GLOBAL	Displays global pages currently in the working set of the process.
/WORKING_SET_LIST = LOCKED	Displays pages locked into the process's working set.
/WORKING_SET_LIST = MODIFIED	Displays working set list entries marked modified.
/WORKING_SET_LIST = PPT	Displays process page table pages.
/WORKING_SET_LIST = PROCESS	Displays process private pages.

For more detailed information, see the *OpenVMS Alpha System Dump Analyzer Utility Manual*.

4.14.3 Arithmetic and Logical Operators

There are two new SDA unary operators. Table 4–25 shows the operators and their meanings.

Table 4–25 New Unary Operators

Unary Operator	Meaning
^P	When used with the unary operator @, it specifies a physical address.
^V	When used with the unary operator @, it specifies a virtual address.

For more detailed information, see the *OpenVMS Alpha System Dump Analyzer Utility Manual*.

4.14.4 Syntax for Displaying Range of Page Table Entries

The new syntax for displaying the range of page table entries by using the SHOW PAGE_TABLE command or the SHOW PROCESS/PAGE command. Table 4–26 shows the command syntax and its meaning.

Table 4–26 New Range Page Table Syntax

Symbol	Meaning
m:*n*	Displays the page table entries that correspond to the range of virtual addresses from *m* to *n*.
m;*n*	Displays the page table entries that correspond to the range of *n* bytes starting at virtual address *m*.

For more detailed information, see the *OpenVMS Alpha System Dump Analyzer Utility Manual*.

4.14.5 Behavior for the READ Command

If no device or directory is given in the file specification, and the file specification is not found in SYS$DISK:[default_dir], then SDA attempts to open the file SDA$READ_DIR:filename.type. If no type has been given in **filespec**, SDA first tries .STB and then .EXE.

The SDA$READ_DIR logical name is defined in the system locial name table. You can use this logical name to redirect SDA to the SYS$SYSTEM area or private area.

For more detailed information, see the *OpenVMS Alpha System Dump Analyzer Utility Manual*.

4.14.6 New DUMPSTYLE File Features

OpenVMS Alpha allows you to write the system dump file to a device other than the system disk. This is useful in large-memory systems and in clusters with common system disks where sufficient disk space, on one disk, is not always available to support your dump file requirements. To perform this activity, the DUMPSTYLE system parameter must be correctly enabled to allow the bugcheck code to write the system dump file to an alternate device.

The requirements for writing the system dump file off the system disk (DOSD) are the following:

- The dump device directory structure must resemble the current system disk structure. The [SYSn.SYSEXE]SYSDUMP.DMP file will reside there, with the same system root.

  You can use AUTOGEN to create this file. In the MODPARAMS.DAT file, the following symbol prompts AUTOGEN to create the file:

  ```
  DUMPFILE_DEVICE = $nnn$ddcuuuu
  ```

- The dump device cannot be part of a volume set.
- The dump device cannot be part of a shadow set unless the dump device is also the system disk and is a master member of the shadow set.
- The DUMP_DEV environment variable must exist on your system. You specify the dump device at the console prompt, using the following format:

  ```
  >>> SET DUMP_DEV device-name[...]
  ```

  On some CPU types, you can enter a list of devices. The list can include various alternate paths to the system disk and the dump disk.

By specifying an alternate path DUMP_DEV, the disk can fail over to the alternate path when the system is running. If the system crashes subsequently, the bugcheck code can use the alternate path by referring to the contents of DUMP_DEV.

When you enter a list of devices, however, the system disk must come last.

For information on how to write the system dump file to an alternative device to the system disk, see the *OpenVMS System Manager's Manual: Tuning, Monitoring, and Complex Systems*.

4.15 System Services Features

The following sections describe the new system services features included in OpenVMS Version 7.1.

4.15.1 Changed Services: Enhanced to Support 64-Bit Addressing (Alpha Only)

In OpenVMS Alpha Version 7.1, the following system services have been enhanced to support 64-bit addressing:

$ASCTIM
$ASCUTC
$BINTIM
$BINUTC
$CRELNM
$CRELNT
$DELLNM
$GETJPI
$GETJPIW
$GETSYI
$GETSYIW
$GETUTC
$IDTOASC
$NUMTIM
$NUMUTC
$PROCESS_SCAN
$SETIME
$TIMCON
$TRNLNM

For complete information, see *OpenVMS System Services Reference Manual*.

4.15.2 New System Services

The following sections describe system services that are new with OpenVMS Version 7.1. These services enable you to tailor OpenVMS scheduling to meet your local requirements. For more information about these new services, see the *OpenVMS System Services Reference Manual*.

4.15.2.1 $AVOID_PREEMPT

The Avoid Process Preemption service requests that the scheduler avoid preempting a process or thread.

While the OpenVMS operating system provides many services for interlocking resources between processes, they are fairly heavyweight in that they require going into kernel mode to dispatch and operate. Some high-performance applications have implemented lightweight user-mode interlocks for some of

their shareable resources. However, the performance advantage of avoiding the transition into kernel mode can be lost if the process that holds this interlock is scheduled off of the CPU, preventing other processes from executing until the preempted process or thread runs again and releases the interlock.

The $AVOID_PREEMPT system service provides a lightweight, caller's-mode method to request that the OpenVMS scheduler avoid preempting this process if at all possible.

4.15.2.2 $SETUP_AVOID_PREEMPT

The Setup for Process Preemption Avoidance service is a kernel-mode initialization routine that locks the necessary internal data structures in memory so scheduling routines can access them. A process or thread can then set or clear the indicator bit by calling $AVOID_PREEMPT.

4.15.2.3 $FREE_USER_CAPABILITY

The Release a Reserved User Capability service releases a user capability back to the global pool and indicates to other processes that the resource is available.

This service can be used along with the $GET_USER_CAPABILITY, $CPU_CAPABILITIES, and $PROCESS_CAPABILITIES system services to coordinate the sharing of global system resources by local processes.

4.15.2.4 $GET_USER_CAPABILITY

The Reserve a User Capability service provides a way for discrete processes to communicate and synchronize their use of a user capability in the system. This service can indicate whether a particular user capability has been reserved and can return the current reservation state of all user capabilities in the system.

5
Documentation Features

This chapter describes documentation features of interest to OpenVMS users.

5.1 Available Documentation for OpenVMS Version 7.1

For OpenVMS Version 7.1, documentation is offered in a variety of formats:

- The entire OpenVMS documentation set has been revised and is available in printed form.
- Several OpenVMS Version 7.1 manuals are available on the OpenVMS Documentation CD–ROM in HTML — a format that is readable with the Netscape Navigator software and other web browsers. For more information, see Section 5.2.
- The entire OpenVMS documentation set is available in Bookreader format on the OpenVMS Version 7.1 Documentation CD–ROM.
- A few manuals, such as the Release Notes and Installation Guides, are available in PostScript and ASCII text format.

5.2 Select Documentation Offered in HTML Format

For OpenVMS Version 7.1, a selected group of manuals is offered in HTML format. These manuals are stored in the [HTML] directory on the OpenVMS Version 7.1 Documentation CD–ROM.

Use the INDEX.HTM Web page located in the [HTML] directory on the Documentation CD–ROM to browse the books. You can view these files with any Level 2 Internet browser, but Digital suggests using the Netscape Navigator Version 2.01 software. If needed, download the Netscape Navigator from the Internet Product Suite CD–ROM, which is part of your OpenVMS Version 7.1 package.

5.3 DECnet-Plus for Version 7.1 Documentation

DECnet-Plus, formerly named DECnet/OSI, has replaced DECnet for OpenVMS Phase IV in the main operating system installation menu. To assist customers choosing to move from DECnet Phase IV to DECnet-Plus, OpenVMS is delivering a one-time complimentary offering that includes DECnet-Plus binaries and a DECnet Plus Starter Kit. Depending on the OpenVMS offering, the DECnet-Plus Starter Kit documentation is provided in printed or online format.

The DECnet-Plus Starter Kit for OpenVMS Version 7.1 customers includes the following documents:

- *DECnet-Plus for OpenVMS Installation and Basic Configuration*
- *DECnet-Plus for OpenVMS Applications Installation and Advanced Configuration Manual*

- *DECnet-Plus for OpenVMS Network Management*
- *DECnet-Plus Planning Guide*
- *DECnet-Plus Problem Solving*
- *DECnet-Plus Network Control Language Reference*
- *DECnet-Plus for OpenVMS Introduction and User's Guide*
- *DECnet-Plus for OpenVMS Release Notes*
- *DECnet-Plus for OpenVMS Installation Quick Reference Card*
- *DECnet-Plus for OpenVMS Network Management Quick Reference Guide*

Additional DECnet-Plus manuals are available on the OpenVMS Documentation Version 7.1 CD–ROM in Bookreader format or can be ordered in printed form through DECdirect (800-344-4825). Refer to the *DECnet-Plus for OpenVMS Introduction and User's Guide* for further information on DECnet-Plus for OpenVMS manuals. In future releases, DECnet-Plus manuals will not be part of the OpenVMS Documentation Sets and must be purchased separately.

DECnet for OpenVMS Phase IV binaries will ship on the OpenVMS Version 7.1 operating system media. The DECnet for OpenVMS manuals are no longer part of the OpenVMS Documentation Set but are included on the OpenVMS Documentation CD–ROM in DECW$BOOK and PostScript format. You can also order the printed books separately through DECdirect (800-344-4825). Table 5–1 lists the DECnet Phase IV manuals.

Table 5–1 DECnet Phase IV Manuals

Manual	Order Number
DECnet for OpenVMS Guide to Networking	AA–PV5ZA–TK
DECnet for OpenVMS Networking Manual	AA–PV60A–TK
DECnet for OpenVMS Network Management Utilities	AA–PV61A–TK

5.4 Device Support Documentation

OpenVMS device support documentation has been reorganized for OpenVMS Version 7.1. Two books about writing OpenVMS Alpha device drivers are now available, and several device support books from previous releases have been archived.

The OpenVMS Alpha device driver books available as of OpenVMS Version 7.1 are as follows:

- *Writing OpenVMS Alpha Device Drivers in C* describes how to write device drivers in the C programming language for devices connected to Alpha processors.
- *Creating an OpenVMS Alpha Device Driver from an OpenVMS VAX Device Driver* explains how to change OpenVMS VAX device driver code to be compiled, linked, loaded, and run on OpenVMS Alpha systems. This book also describes OpenVMS Alpha driver entry point routines, system routines, and macros for the MACRO-32 compiler.

Table 5–2 summarizes the Alpha and VAX device support documentation changes for OpenVMS Version 7.1. It lists the device support books that have been archived, and indicates whether the information they contain is available in other books.

Table 5–2 OpenVMS Version 7.1 Device Support Books

Title	New Location in Version 7.1
Creating an OpenVMS AXP Step 2 Device Driver from a Step 1 Device Driver	Same
Creating an OpenVMS AXP Step 2 Device Driver from an OpenVMS VAX Device Driver	*Creating an OpenVMS Alpha Device Driver from an OpenVMS VAX Device Driver* (This book now includes reference information.)
OpenVMS Alpha Device Support: Developer's Guide	*Writing OpenVMS Alpha Device Drivers in C*
OpenVMS Alpha Device Support: Reference	C reference information is available in *Writing OpenVMS Alpha Device Drivers in C*. MACRO-32 reference information is included in the *Creating an OpenVMS Alpha Device Driver from an OpenVMS VAX Device Driver* book.
OpenVMS VAX Device Support Manual	Same
OpenVMS VAX Device Support Reference Manual	Same

5.5 Migration Documentation

OpenVMS migration documentation is designed to help developers and system managers move computing environments and applications from an OpenVMS VAX to an OpenVMS Alpha system.

For Version 7.1, migration information is located in *Migrating an Application from OpenVMS VAX to OpenVMS Alpha* and *Porting VAX MACRO Code to OpenVMS Alpha*. The following documents have been archived. Table 5–3 lists the order numbers for these documents.

- *A Comparison of System Management on OpenVMS AXP and OpenVMS VAX*
- *Migrating an Environment from OpenVMS VAX to OpenVMS Alpha*
- *Migrating to an OpenVMS AXP System: Planning for Migration*
- *Migrating to an OpenVMS AXP System: Recompiling and Relinking Applications*
- *OpenVMS Compatibility Between VAX and Alpha*

5.6 Archived Documentation

Several books have been archived for Version 7.1. Much of the information in these books has been incorporated in other manuals or updated in online Help. Archived books are available in PostScript and Bookreader format on the Documentation CD–ROM, or you can order a printed book through DECdirect at 800-344-4825. Table 5–3 lists the books archived for this release.

Documentation Features
5.6 Archived Documentation

Table 5–3 Manuals Archived for Version 7.1

Manual	Order Number
A Comparison of System Management on OpenVMS AXP and OpenVMS VAX	AA–PV71B–TE
Building Dependable Systems: The OpenVMS Approach	AA–PV5YB–TE
Creating an OpenVMS AXP Step 2 Device Driver from a Step 1 Device Driver	AA–Q28TA–TE
Creating an OpenVMS AXP Step 2 Device Driver from an OpenVMS VAX Device Driver	AA–Q28UA–TE
Guide to OpenVMS AXP Performance Management	AA–Q28WA–TE
Guide to OpenVMS Performance Management	AA–PV5XA–TE
Migrating an Environment from OpenVMS VAX to OpenVMS Alpha	AA–QSBLA–TE
Migrating to an OpenVMS AXP System: Planning for Migration	AA–PV62A–TE
Migrating to an OpenVMS AXP System: Recompiling and Relinking Applications	AA–PV63A–TE
OpenVMS Alpha Device Support: Developer's Guide	AA–Q28SA–TE
OpenVMS Alpha Device Support: Reference	AA–Q28PA–TE
OpenVMS Compatibility Between VAX and Alpha	AA–PYQ4C–TE
OpenVMS Glossary	AA–PV5UA–TK
OpenVMS Programming Environment Manual	AA–PV66B–TK
OpenVMS Software Overview	AA–PVXHB–TE
OpenVMS VAX Device Support Manual	AA–PWC8A–TE
OpenVMS VAX Device Support Reference Manual	AA–PWC9A–TE

For more detailed information on OpenVMS documentation including components and part numbers, see the *Overview of OpenVMS Documentation*.

6

Automatically Configuring Devices for OpenVMS Alpha

File-based autoconfiguration is a new feature that enables OpenVMS Alpha to configure third-party hardware devices automatically. As of OpenVMS Alpha Version 7.1, device configuration tables are constructed from ASCII text files on the OpenVMS Alpha operating system disk. By adding simple descriptions of their devices in the appropriate ASCII text file, third parties and end users can configure non-Digital supported devices and load user-written device drivers.

This chapter briefly explains device configuration and describes how to use the new file-based autoconfiguration method to configure non-Digital supported devices.

6.1 Overview of Device Configuration

A device is configured on OpenVMS when system code is able to locate it on a bus, give it a name, and load a device driver for it. When a device is autoconfigured, all these steps happen without any user intervention.

Autoconfiguration is the process of discovering the hardware devices on a system and loading the appropriate device drivers. OpenVMS discovers devices during the boot process in a bus-specific manner. The discovery process includes storing data about detected devices in bus-specific data structures. Later, these data structures are used to search configuration tables of known devices. The configuration table provides the information necessary to determine the driver name, device name, and other parameters for loading and connecting the appropriate driver.

Prior to OpenVMS Alpha Version 7.1, configuration tables were built into the OpenVMS kernel and could not be modified without replacing a system image. As of OpenVMS Alpha Version 7.1, the configuration tables are constructed from ASCII text files on the system disk. A system file (**SYS$SYSTEM:SYS$CONFIG.DAT**) is provided for all OpenVMS supported devices, and a user file (**SYS$SYSTEM:SYS$USER_CONFIG.DAT**) is provided for all third party, layered product, and user-written device drivers. These files are read by the system during the boot process and used to create a set of configuration tables. These tables are used for subsequent autoconfiguration of hardware devices. Although the tables are built from two files and collected by bus type, they may be considered one logical configuration table of known devices.

Section 6.2 describes how to use the SYS$SYSTEM:SYS$USER_CONFIG.DAT file to autoconfigure devices.

6.2 Using File-Based Autoconfiguration

File-based autoconfiguration reads two files during system boot to build the configuration tables of known devices:

SYS$SYSTEM:SYS$USER_CONFIG.DAT
SYS$SYSTEM:SYS$CONFIG.DAT

Both files use the same format, and the data from both files are combined to create the configuration tables for each bus on the system. The SYS$USER_CONFIG.DAT file is read first, and takes precedence over any duplicate device descriptions contained in both files. If multiple device descriptions exist in a single file, the first occurrence of the description is used.

The configuration files consist of device description blocks, each of which provides the information needed to configure the correct device driver for a device.

Each device description block consists of a series of statements starting with a DEVICE keyword, and ending with the END_DEVICE keyword. Between these two keywords, additional keywords define the hardware ID, the device name, the driver name, the bus type, and any other required or optional information.

The SYS$USER_CONFIG.DAT file is an ASCII text file, which can be processed with any utility that handles variable length record files (for example, a text editor or DCL commands).

Note

The SYS$CONFIG.DAT file is read-only and should NOT be modified by users or by third parties. It should only be modified by Digital, and it may be replaced by OpenVMS upgrades. Inappropriate edits to this file could result in the inability to boot the system.

6.2.1 Adding Descriptions to USER_CONFIG.DAT

Statements in the SYS$SYSTEM:SYS$USER_CONFIG.DAT take the general form:

```
KEYWORD = value
```

Where the value is a string, a quoted string, or a numeric value. The END_DEVICE keyword has no associated value. For example, a minimal description of a non-disk device might look like the following:

```
DEVICE          = "My device"
 NAME           = UU
 DRIVER         = USER$UUDRIVER
 ID             = 0x0005111
 ADAPTER        = PCI
 FLAGS          = NOVECTOR
END_DEVICE
```

In this example, the description indicates that when a device with the hardware ID of 5111 (hex) is found on a PCI bus, the device named UU should be configured, and the USER$UUDRIVER device driver should be loaded. This example also specifies that the driver does not want to be connected to an interrupt vector. (If the bus information had a vector, it would be ignored.)

In addition to the values shown in the previous example, the following implied values can also be specified:

```
UNITS           = 1
NUM_VECTORS     = 1
```

These values usually default to the correct values for a single unit controller.

6.2.2 Configuration File Syntax

The following syntax rules apply to the SYS$USER_CONFIG.DAT file.

- All white space is equivalent to a single space. Blank lines, multiple spaces, and tabs may be used for clarity, but they are treated by the parser as a single space.
- All strings are converted to uppercase unless within quotes.
- A quotation mark cannot be passed within a string (it will be deleted).
- All numbers are in the C language format: the default is decimal, octal may be specified by providing a leading ‘‘0’’, hex by providing a leading "0x".
- Comments can occur anywhere in the file and can begin with an exclamation point (!). All text on the line from the exclamation point to the end of the line is ignored.

6.2.3 Device Descriptions

A minimal device description consists of DEVICE, NAME, DRIVER, ADAPTER, and END_DEVICE statements. The following keywords are defined for file-based autoconfiguration device descriptions:

- **DEVICE** = string

  The DEVICE keyword starts a device description. The DEVICE keyword takes a string argument, which is an arbitrary description of the device being defined.

  Example: DEVICE = "NI (Tulip)"

  The **string** argument is used by utilities like the CLUE CONFIG command in $ANALYZE/SYSTEM. It should be kept short enough to look correct.

```
SDA> CLUE CONFIG
 .
 .
 .
Adapter Configuration:
----------------------
TR Adapter     ADP      Hose Bus   BusArrayEntry  Node Device Name / HW-Id
-- ----------- -------- ---- -------------------- ---- ------------------
 .
 .
 .
 4 PCI         80949900   60 PCI
                               80949B50          1 MERCURY
                               80949BC0  GQA:    3 S3 Trio32/64
                               80949C30  EWA:    5 NI (Tulip)
```

- **END_DEVICE**

  The END_DEVICE terminates the current device description, and causes the parser to create a new table entry if all required keywords have been supplied. No parameters are required.

- **ID** = {number {, number} | string}

  The ID keyword specifies the hardware ID of the device. The hardware ID is a 64-bit quantity and can be specified by a pair of 32-bit values (low-order, high-order), or can be specified as an ASCII string of up to 8 characters. If only one numeric value is provided, the high-order 32-bits are set to zero.

  Example: ID = 0x00051000 ! HW ID of NCR810 in hex

  Example: ID = FLOPPY ! HW ID of Floppy in ASCII

  Example: ID = 0x906010B5, 0x113310DF ! 64-bit ID => low-32, high-32

  The number of bits used by the system depends on the adapter type. EISA and PCI use 32 bits. ISA uses 64 bits. PCI can be extended to 64 bits by using the FLAGS=EXTENDED_ID keyword.

- **NAME** = string

  The NAME keyword specifies the mnemonic for the device. This is usually a two character name, but it can be more.

  Example: NAME = PK

- **DRIVER** = string

  The DRIVER keyword specifies the file name of the device driver for the device.

  Example: DRIVER = SYS$PKEDRIVER

- **ADAPTER** = string

  The ADAPTER keyword indicates the bus type of the device. The current adapters are: *PCI, EISA, ISA, TC (TURBOchannel).*

  Example: ADAPTER = PCI

- **UNITS** = number

  The UNITS keyword indicates the number of units (UCBs) that are created for the controller. The default if not specified is 1.

  Example: UNITS = 1

- **FLAGS** = string

  The FLAGS keyword shows optional information about how to load the driver. More than one flag may be specified, separated by the ‘‘,’’ character.

  Example: FLAGS = NOVECTOR, CASE_BLIND, EXTENDED_ID, ISA_ON_ EISA

NOVECTOR	Device has no interrupt vector.
CASE_BLIND	Causes the ID to be treated as an ASCII string and converted to uppercase. In addition, the ID from the hardware will also be converted to uppercase before comparing. This flag is useful for ISA device strings, because the console ISACFG command will not convert the HANDLE to uppercase. If this flag is specified, the ID must be entered in the description as a string, and not as a number, or an error will be displayed and the description ignored.

EXTENDED_ID — The flag is valid only for devices on the PCI bus. Normally, only 32-bit hardware IDs are used for PCI. If you have a PCI board that supports the V2.1 PCI bus specification, set this flag. The upper 32-bits of the hardware ID should contain the subsystem ID and the subsystem vendor ID. For example, in the following example, 1133 is the subsystem ID, and 10DF is the subsystem vendor ID.

```
FLAGS = EXTENDED_ID
 ID = 0x906D10B5, 0 x 113310DF
```

If the EXTENDED_ID flag will be used to load a special driver, but the same ID without the EXTENDED_ID bit will be defined to load a generic driver, the description with the EXTENDED_ ID should be located before the generic description.

ISA_ON_EISA — This flag is valid only when the value EISA is given to the ADAPTER keyword, and the device is an ISA device. ISA devices should set this flag to insure that the value given to the ID keyword is interpreted correctly. For ISA devices, the system keeps the ID value as an ASCII string. When the system believes that this is an EISA device, the ID value is compressed into binary format.

- **PRIVATE_DATA** = string

  This keyword is only supported when the value "ISA" is given to the ADAPTER keyword. The statement may not span lines. If the string contains more than one word, the string should be in quotes. The quotes are not passed as part of the string. An ISA driver may retrieve this string by calling IOC$NODE_DATA and passing the function code IOC$K_ISA_USER_ PARAM.

- **BEGIN_PRIVATE**

  This keyword is only supported when the value "ISA" is given to the ADAPTER keyword. The BEGIN_PRIVATE and END_PRIVATE keywords can be used instead of PRIVATE_DATA. They are not used as part of a PRIVATE_DATA statement. When using BEGIN_PRIVATE and END_ PRIVATE, all data contained between the keywords is passed to the driver. The data may contain special characters like quotation marks. The data may span multiple lines. When multiple lines are used, the driver will find a line-feed character to indicate a new line.

  BEGIN_PRIVATE and END_PRIVATE must be used when trying to convert some entries in ISA_CONFIG.DAT to file based autoconfiguration. For example, the statement in ISA_CONFIG.DAT

  ```
  USER_PARAM = "some data"
  ```

  must be converted to

  ```
  BEGIN_PRIVATE
  "some data"
  END_PRIVATE
  ```

  The USER_PARAM keyword in ISA_CONFIG.DAT passes the quotation marks, so PRIVATE_DATA cannot be used to do the conversion.

  An ISA driver can retrieve the data between BEGIN_PRIVATE and END_ PRIVATE by calling IOC$NODE_DATA with the function code IOC$K_ISA_ USER_PARAM.

- **END_PRIVATE**

  The END_PRIVATE keyword can only be used to terminate data that is added after a BEGIN_PRIVATE statement.

- **NUM_VECTORS** = number

  The NUM_VECTORS keyword specifies the number of vectors that the device uses. The default if not specified is 1.

  Example: NUM_VECTORS = 4

Table 6–1 indicates keyword that can be included in the configuration file.

Table 6–1 Keyword Summary

Keyword	Required	Description
DEVICE	Yes	Begins a device description
END_DEVICE	Yes	Ends a device description
ID	Yes	Specifies the hardware ID.
NAME	Yes	Device Name
DRIVER	Yes	Driver name
ADAPTER	Yes	Adapter type
UNITS	No	Units *Default: 1*
FLAGS	No	Device Flags: *Default: No flags*
PRIVATE_DATA	No	Specifies private data
BEGIN_PRIVATE	No	Specifies start of private data
END_PRIVATE	No	Specifies end of private data
NUM_VECTORS	No	Number of Vectors *Default: 1*

6.2.4 Rebuilding the USER_CONFIG.DAT File

The REBUILD keyword requests SYSMAN to rebuild the configuration tables attached to each of the adapter blocks by re-reading and parsing SYS$SYSTEM:SYS$USER_CONFIG.DAT and SYS$SYSTEM:SYS$CONFIG.DAT. The REBUILD command will always rebuild all of the configuration tables, regardless of the type of bus. It is then necessary to re-execute the AUTOCONFIGURE command to load the device drivers for any newly defined devices:

```
$ MC SYSMAN IO REBUILD
$ MC SYSMAN IO AUTOCONFIGURE
```

Note that once a driver has been loaded for a device, it cannot be reloaded.

The MC SYSMAN IO REBUILD/VERIFY command causes SYSMAN to read and process the SYS$SYSTEM:SYS$USER_CONFIG.DAT and SYS$SYSTEM:CONFIG.DAT files, but not to rebuild the configuration files for OpenVMS. Messages will be displayed for any errors that are encountered. This command can be used by developers to test new changes to SYS$SYSTEM:SYS$USER_CONFIG.DAT without modifying the current system.

6.3 Supported Buses for User Devices

File-based autoconfiguration is supported for user-written device drivers on PCI, ISA, EISA, and TURBOchannel buses. This section includes additional information specific to configurations.

6.3.1 ISA Configuration

ISA devices do not provide a readable device ID that can be discovered during bus probing. A user must explicitly indicate the presence of the device at the console and must also reserve resources for the device at the console (IRQs, I/O ports, etc.). Once the device is known to the console OpenVMS can then autoconfigure it using file-based autoconfiguration.

ISA devices may be used on either an ISA bus or an EISA bus. If the system has an ISA bus, the device is configured at the console using ISACFG. If the system has an EISA bus, the ISA device is configured using ECU. Both console utilities allow the users to reserve device resources.

6.3.1.1 Configuring ISA Devices on the ISA Bus

In previous versions of OpenVMS Alpha, ISA devices on an ISA bus required an entry in the SYS$MANAGER:ISA_CONFIG.DAT file that defined the hardware and the use of the console command ISACFG to reserve system resources like IRQs.

Warning

As of OpenVMS Alpha Version 7.2, the ISA_CONFIG.DAT file will be removed and will no longer be supported. Refer to Section 6.4 for more information.

6.3.1.2 Configuring ISA devices on the EISA Bus

ISA devices must be manually configured using the EISA Configuration Utility (ECU) which is run from the console. Devices must have a CFG file provided on a DOS floppy. The CFG file provides a string (up to 7 characters) as the device ID.

See your card manufacturer, or the EISA Bus Specification for details on CFG file format.

The ECU floppy (DOS format) contains an example ISA CFG file (ISA000.CFG) that may be used as a model for new configuration files. For more information, refer to the EISA Bus Support chapter in *Writing OpenVMS Alpha Device Drivers in C*.

Once the ECU has been run, the device can be configured using file-based autoconfiguration.

Note

ISA devices cannot be easily autoconfigured on EISA bus systems on versions prior to OpenVMS Alpha Version 7.1 because the ID is not copied from the ECU data into the OpenVMS bus structures.

6.4 SYS$MANAGER:ISA_CONFIG.DAT Unsupported in Future Releases

Support for using the SYS$MANAGER:ISA_CONFIG.DAT file to configure ISA devices will be discontinued in a future release of OpenVMS Alpha. If you use this file, you should convert to using the ISACFG utility from the console and the file-based autoconfiguration method described in this chapter.

Table 6–2 contains a list of keywords from ISA_CONFIG.DAT and their equivalents in either file-based autoconfiguration or the ISACFG utility.

Table 6–2 ISA_CONFIG.DAT Keywords and Equivalents

ISA_ CONFIG.DAT	File-based Autoconfigure	ISACFG
Not used	ID	-handle
NAME	NAME	
DRIVER	DRIVER	
IRQ		irq*x*
NODE		slot
DMA		dmachan*x*
PORT		iobase*x*
MEM		membase*x*
FLAGS	Bit 1 (unsupported)	
	Bit 2 (FLAG=NOVECTOR)	
USER_PARAM	PRIVATE_DATA	

An entry in ISA_CONFIG.DAT is matched to internal data kept for an ISA device using the number specified with the NODE keyword. But when you use SYS$USER_CONFIG.DAT file to configure an ISA device, the ID keyword is used to match the block, which defines the device, to data entered from the console with the ISACFG command. The value give to the ID keyword must be the same as the value specified with the ISACFG -handle keyword.

Any identification string can be used for an ISA device. It should be eight characters or less. The ISACFG command does not set the -handle value to upper case, so two methods can be used to force the value to match one specified using the configuration keyword ID. You can specify the ID value in the correct case inside of quotation marks (matching the case you used for the -handle value). Or you can use the configuration keyword FLAGS=CASE_BLIND, which will cause a blind comparison to be done.

For example, if you use the following in ISACFG:

```
>>>isacfg -slot 3 -dev 0 -mk -enadev 1 -type 1 -handle MyDevice
```

you can match that to the following entry in SYS$USER_CONFIG.DAT:

```
DEVICE   = "My Device"
ID = MYDEVICE
FLAGS = CASE_BLIND
            .
            .
            .
END_DEVICE
```

Descriptions of the conversion for each parameter in ISA_CONFIG.DAT are as follows:

NAME = xx
Use the NAME keyword in SYS$USER_CONFIG.DAT. Use the same value, where xx is the device code. (The device code is usually 2 letters.)

Example: NAME = ER

DRIVER = driver_name
Use the DRIVER keyword in SYS$USER_CONFIG.DAT. Use the same value for file-based autoconfiguration. driver_name is the name of the driver in SYS$LOADABLE_IMAGES.

Example: DRIVER = SYS$ERDRIVER

IRQ = i
Use irq*x* in the ISACFG utility. You can express four irqs: -irq0 thru -irq3. Use the same value as you used for ISA_CONFIG.DAT. The IRQ is a value from 0 to 15, which specifies which ISA IRQ the device uses to report interrupts.

Example:

This example assigns IRQs 10 and 5 to the device.

```
>>>isacfg -slot 3 -dev 0 -mk -handle MYDEV -enadev 1 -etyp 1  -irq0 10 -irq1 5
```

NODE = n
Use -slot in the ISACFG utility. The slot number (n) does not represent the slot in which the device resides. It is a logical, not a physical number. However, the number does have to be between 1 and the maximum number of slots on the machine. Slot number 0 is not available to users.

Example:

```
>>>isacfg -slot 3 -dev 0 -mk -enadev 1 -etyp 1 -handle MYDEV -dmachan0 1 -irq0 10
```

This example assigned values to a device represented by slot 3. There must be at least 3 slots on the machine on which this command is executed. To see which logical slots are being used, enter the following command:

```
>>>isacfg -all
```

DMA = (j,k, ...)
Use -dmachan*x* in the ISACFG utility. Values j,k etc. are values zero through seven, which specify the channels of the DMA controller which the device is using to relay information. Use the same values for j, k, etc. with ISACFG, but assign each one to a different DMA channel. You can specify four DMA channels, using the keywords -dmachan0 thru -dmachan3.

Example:

```
>>>isacfg -slot 3 -dev 0 -mk -enadev 1 -etyp 1 -handle MYDEV -irq0 10 -dmachan0 1 -dmachan1 3
```

This example assigned two dma channels, 1 and 3, to the device.

PORT = (aa:b, cc:d, ...)
Use -iobase*x* in the ISACFG utility. You can specify six ports using the keywords -iobase0 through -iobase5. There is no equivalent for the length fields *b, d* etc. The ISACFG utility assumes that the driver knows the length of the port. Drivers that called the IOC$NODE_DATA routine with the keyword IOC$K_ EISA_IO_PORT to obtain the length in the upper word of the returned longword should stop examining the upper word. With ISA_CONFIG.DAT, the length was returned; but with ISACFG, the length is always 8.

Example:

```
>>>isacfg -slot 3 -dev 0 -mk -enadev 1 -etyp 1 -handle AAA321 -irq0 10 -iobase0 2F8
```

This example assigned port 2F8 to the device.

MEM = (ee:f, gg:h, ...)
Use the ISACFG keywords -membase*x*, to specify the memory base, and -memlen*x*, to specify the memory length. Use the same values for ee, gg etc. and f, h etc. as you used for ISA_CONFIG.DAT. You can specify three memory regions using the keywords membase0 through membase2 and memlen0 through memlen2.

Example:

```
>>>isacfg -slot 3 -dev 0 -mk -enadev 1 -etyp 1 -handle MYDEV -irq0 10 -membase0 80000 -memlen0 20
```

FLAGS = n
The FLAGS field in ISA_CONFIG.DAT had 2 meaningful bits:

Bit 0 indicates the device being configured is a SCSI adapter.

Bit 1 indicates that no interrupt is required for the device.

Because it was never possible to use bit 0, it is not currently supported in file-based autoconfiguration. Bit 1 can be expressed with the file-based autoconfiguration FLAGS=NOVECTOR statement.

USER_PARAM = text
Use the PRIVATE_DATA keyword in file-based autoconfiguration to represent this value. If you used quotation marks with the USER_PARAM value, you must use BEGIN_PRIVATE and END_PRIVATE to continue to pass the quotation marks to the driver. For ISA devices, the PRIVATE_DATA values can be obtained the same way as USER_PARAM (that is, by using the IOC$NODE_DATA routine with the IOC$K_ISA_USER_PARAM keyword).

While using ISACFG you must also be familiar with the following commands:

To return the configuration to its initial state:

```
>>>isacfg -init
```

To save your changes:

```
>>>init
```

To delete an entry:

```
>>>isacfg -slot 1 -dev 0 -rm
```

To see all the devices currently configured:

```
>>>isacfg -all
```

To modify a device, use -mod:

```
>>>isacfg -slot 2 -dev 0 -mod  (etc.)
```

The following keywords do not have equivalents in ISA_CONFIG.DAT:

-enadev a_number	Takes the numbers 0 (disabled) and 1 (enabled). It allows you to disable a device so that it will not be used in resource allocation calculations.
-etyp a_number	Defines an entry type for this entry. OpenVMS supports only the values 0 and 1. It should always be specified as a 1. It takes the following values: 0 Causes the entry to be deleted 1 Single option 2 Embedded multiport device 3 Multiport option device

6.5 Error Information

This section describes system messages that might appear when you are using file-based autoconfiguration to configure a device.

For each message, the text of the message appears in uppercase letters and is followed by an explanation of why the message occurred and, where applicable, what you should do to correct the problem.

%READ_CONFIG-W-BAD_HWID, the HW ID string cannot be zero. Device *device_name* ignored at line *line_number*.

Explanation: A explicit attempt was made to make the ID zero. The ID cannot be zero for a hardware device. The device description will not be added.

%READ_CONFIG-W-BAD_NUMBER, the argument *text string* to the *keyword* command is not a valid number. Device *device_name* ignored at line *line_number*.

Explanation: The numeric value in an argument was illegal. The device description will not be added.

%READ_CONFIG-W-BAD_EISA_ID, the ID *text string* must be 3 alpha characters, followed by 4 hex digits. Device *device_name* ignored at line *line_number*.

Explanation: EISA device IDs are of the form XXXnnnv where XXX are three uppercase ASCII alphabetic characters, and nnn is a 3 digit hex product ID, and v is a hex revision value. When a EISA ID is entered as a string, it is validated, and compressed into a 32-bit binary ID that can be compared to the ID read from the EISA card.

User Action: To enter this value directly, use a numeric ID.

If the device is an ISA device on the EISA bus, use the flag ISA_ON_EISA to prevent compression of the ID.

In OpenVMS Alpha Version 7.1, the device description will not be added.

%READ_CONFIG-W-UNKOWN_COMMAND, unknown command: *keyword* at line *line_number*.

Explanation: An unknown command was found. It is ignored.

Automatically Configuring Devices for OpenVMS Alpha 6.5 Error Information

%READ_CONFIG-W-MISSING_INFO, a required field was not provided for: DEVICE = *text string*, NAME = *text string*, or DRIVER = *text string* at line *line_number*.

Explanation: One of the listed fields is blank, and is required. The device description will not be added.

%READ_CONFIG-W-NOT_ADDED, a error was detected for DEVICE = *text string*, NAME = *text string*, or DRIVER = *text string*, not added.

Explanation: An error occured in the description (and is displayed prior to this message). This message confirms that the device description will not be added.

%READ_CONFIG-W-MISSING_END_DEVICE, missing END_DEVICE. Device *text string* ignored at line *line_number*.

Explanation: A DEVICE command was encountered before a END_DEVICE was found for the current device description. The device description will not be added.

%READ_CONFIG-W-INSFARGS, insufficient arguments in the *keyword* command, needs *number*, found *number* at line *line_number*.

Explanation: Not enough arguments were provided to the command. The device description will not be added.

%READ_CONFIG-W-NOEQUALS, missing equal sign in *keyword* command at line *line_number*.

Explanation: The command expected an equal sign. The device description will not be added.

%READ_CONFIG-W-NOSTRING, string value is NULL in *keyword* command at line *line_number*.

Explanation: A required string argument had a zero length. The device description will not be added.

%READ_CONFIG-W-NEEDVALUE, needed value in *keyword* command, found *text string* instead at line *line_number*.

Explanation: A required numeric value contained an illegal string. The device description will not be added.

%READ_CONFIG-W-BADUNITS, illegal UNITS value *text string* at line *line_number*, forcing UNITS = 1.

Explanation: A negative or null UNITS value was entered. The device description will not be added.

%READ_CONFIG-W-UNKNOWNADAPTER, unknown adapter Type: *text string* at line *line_number*.

Explanation: The ADAPTER type was unknown. The device description will not be added.

%READ_CONFIG-W-MISSINGADAPTER, the device *device_name* has an ID, but no ADAPTER, premature END_DEVICE at line *line_number*.

Explanation: The description requires an ADAPTER, which is missing. The device description will not be added.

%READ_CONFIG-W-UNKNOWN_FLAG, unknown flag: *text string* at line *line_number*.

Explanation: An unknown flag was specified. The device description will not be added.

%READ_CONFIG-W-NULL_HW_ID, no ID was specified for DEVICE *device_name*, premature END_DEVICE at line *line_number*.

Explanation: A hardware ID is required, and is missing. The device description will not be added.

%READ_CONFIG-W-ID_NOT_STRING, the ID for device *device_name* is not a string, but CASE_BLIND is specified, at line *line_number*.

Explanation: The CASE_BLIND flag implies that the device ID is a string, but the ID was specified as a number. The device description will not be added.

%READ_CONFIG-W-LINE_TRUNCATED, more than *number* characters in line, at line *line_number*.

Explanation: Lines cannot exceed 256 characters in length. The device description will not be added.

%READ_CONFIG-W-NOT_IN_DEVICE, *keyword* command encountered with no preceding DEVICE command, at line *line_number*.

Explanation: A valid command was found, but it was not preceded by a valid DEVICE command. The device description will not be added.

%READ_CONFIG-W-NOT_IN_PRIVATE, END_PRIVATE command encountered with no preceding BEGIN_PRIVATE command, at line *line_number*.

Explanation: An END_PRIVATE was found, but there was a missing or invalid BEGIN_PRIVATE command preceding it. The device description will not be added.

%READ_CONFIG-W-TOOMANYARGS, *command* expected *number* arguments, but found *number* at line *line_number*.

Explanation: The command had too many arguments specified. The device description will not be added.

%READ_CONFIG-W-DUPLICATE_ID, Device *device_name* has the same ID as a previous device for this bus, at line *line_number*.

Explanation: A previous device in the file has the same ID and same ADAPTER values. The initial occurance in the file will be used, and the second will never match.

%READ_CONFIG-W-COMMAND_REPEAT, duplicate *keyword* command for device *device_name* is ignored, at line *line_number*.

Explanation: A command was repeated within a description. Only the initial occurance within the description will be used, and the subsequent occurances will be ignored.

%READ_CONFIG-F-NO_SYSTEM_CONFIG, the system file *filename* was not found, your system will not boot.

Explanation: No SYS$CONFIG.DAT file was found in the SYS$COMMON or SYS$SYSROOT directories. It is highly unlikely that the system will boot.

Automatically Configuring Devices for OpenVMS Alpha 6.5 Error Information

%READ_CONFIG-F-NOMEMORY, could not allocate memory, requested *text string* bytes.

Explanation: Non-paged memory could not be allocated for the configuration table. Increase NPAGEDYN in SYSBOOT and retry the operation.

%READ_CONFIG-I-NOAUTOCONFIG, *filename* will not be read, use SYSMAN IO REBUILD comamand to include devices after boot.

Explanation: The NOAUTOCONFIG sysgen parameter will cause the SYS$USER_CONFIG.DAT file to be ignored during the bootstrap. This allows a corrupt file to be bypassed. Once booted, the SYSMAN IO REBUILD command will read the file and add the devices to the system configuration database, and a SYSMAN IO AUTO will then configure any devices from the SYS$USER_CONFIG.DAT file.

A

New OpenVMS System Messages

This release includes new or changed messages for the following OpenVMS facilities:

- BACKUP, Backup Utility
- CLI, Command Language Interpreter (DCL)
- Cluster Port Driver
- LICENSE, License Management Utility
- LOGIN, Login Procedure
- MC, MEMORY CHANNEL
- MOUNT, Mount Utility
- PPPD, Point-to-Point Protocol Utility
- SDA, System Dump Analyzer
- SDP, Set Dump Priority Utility
- SET, SET PASSWORD Command
- SHADOW, Volume Shadowing Software
- SMI, System Management Integrator/Server
- SYSINIT, System Initialization
- SYSMAN, System Management Utility
- SYSTEM, System Services
- UAF, Authorize Utility

This appendix alphabetically lists and describes most messages that have been added or changed for this release. New and changed messages for the following facilities are documented with other messages for those facilities in *OpenVMS System Messages: Companion Guide for Help Message Users*: BACKUP, LOGIN, MOUNT, SHADOW, SYSINIT, and Cluster Port Driver. MEMORY CHANNEL (MC) messages are in the Help Message database only.

You can access online descriptions of new and existing OpenVMS system messages by using the online Help Message utility. 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*.

New OpenVMS System Messages

ABORT, fatal error encountered; operation terminated

Facility: PPPD, Point-to-Point Protocol Utility

Explanation: The fatal error reported in an accompanying message caused the command to fail.

User Action: Correct the problem and retry the operation.

ALPHADUMP, dump file header FLAGS2 DMP$V_ALPHADUMP is set indicating an Alpha dump

Facility: SDA, System Dump Analyzer

Explanation: The dump file you are trying to analyze was written on an OpenVMS Alpha system and cannot be analyzed using the OpenVMS VAX System Dump Analyzer.

User Action: Use the OpenVMS Alpha System Dump Analyzer to analyze this dump file.

ASNCNTINV, unable to show counters on an inactive line

Facility: PPPD, Point-to-Point Protocol Utility

Explanation: The current serial line is not in Point-to-Point Protocol (PPP) mode. No counter information is available for display.

User Action: Switch the serial line into PPP mode using the PPPD command CONNECT [ddcu:], then reissue the PPPD command SHOW/COUNTERS [ddcu:].

ASNSETUP, error with ASN device setup

Facility: PPPD, Point-to-Point Protocol Utility

Explanation: PPPD failed to configure an asynchronous device for the reason described in the accompanying message.

User Action: Take action based on the accompanying message.

ASSIGNCHAN, error trying to open device 'device-name'

Facility: PPPD, Point-to-Point Protocol Utility

Explanation: PPPD failed to access the specified device for the reason described in the accompanying message.

User Action: Take action based on the accompanying message.

AUTHFAIL, authorization failure

Facility: SYSTEM, System Services

Explanation: User authentication failed because of an invalid user name, password, or other account restriction.

User Action: Verify that the user name and password are valid and that account restrictions on the local system and in the external authenticator (if applicable) allow for system access.

BADCHECK, checksum is not valid for 'product' Authorization: 'authorization-number'

Facility: LICENSE, License Management Utility

Explanation: The checksum on the PAK record no longer matches the rest of the data on it.

User Action: Delete the PAK and restore the original one from backup. Either register the original paper PAK or, from a backup copy of the license database, use one of these commands: COPY, MOVE, ISSUE.

BADCHKSUM, dump header checksum incorrect; continuing anyway

Facility: SDA, System Dump Analyzer

Explanation: The header of the dump being analyzed has an incorrect checksum, indicating that the dump has been modified or corrupted. Dump analysis might not be possible.

User Action: No action is required if the cause of the modification or corruption is known, and the dump can still be analyzed. If the problem persists, contact a Digital support representative.

BADIOSBADDR, IOSB buffer address 'address' is not on allocated list

Facility: PPPD, Point-to-Point Protocol Utility

Explanation: The address of the I/O status block (IOSB) is not valid. PPPD tried to use a status block that is not in the IOSB allocation queue.

User Action: Contact a Digital support representative.

BADOVER4GB, dump file contains data written directly from addresses over 4GB

Facility: SDA, System Dump Analyzer

Explanation: The system dump includes data from a physical address above the first 4 gigabytes of system memory. This data cannot be written correctly in all configurations and may cause confusion when the dump is analyzed.

This message indicates that all of the following conditions exist:

- The total system memory exceeds 4 GB.
- Resident libraries have not been installed.
- The boot driver does not support virtual I/O.
- Memory from a physical address above 4GB has been written to the dump.

Under these conditions, a selective memory dump is taken even if a full memory dump is requested. Also, the dump is not compressed even if compression was requested.

User Action: Ensure that resident libraries are used. These are automatically installed when AUTOGEN is used to configure the system.

BADPSBARRAY, invalid value in PSB array; array has been corrupted

Facility: SDA, System Dump Analyzer

Explanation: SDA determined that the PSB array for the specified process contains data that does not conform to design requirements. This status indicates that one or more cells in the array are corrupt.

User Action: Examine the contents of the PSB to determine whether this corruption is responsible for or related to the problems being analyzed.

BADRESSIZ, invalid value for memory reservation size

Facility: SYSMAN, System Management Utility

Explanation: An invalid value was specified for the /SIZE qualifier.

User Action: Specify a positive value for the /SIZE qualifier.

BADSELWGHT, selection weight not valid for 'product'; values range from 0 to 1000

Facility: LICENSE, License Management Utility

Explanation: The selection weight you entered is too large.

User Action: Either specify a smaller value or use the defaults by entering a value of 0.

BATCH, batch operations are not supported at this time

Facility: PPPD, Point-to-Point Protocol Utility

Explanation: An attempt was made to execute a PPPD command in batch mode. Batch mode is not currently supported for PPPD commands.

User Action: Do not attempt to execute PPPD commands in batch mode.

BREAKERR, error sending break on 'device-name'

Facility: PPPD, Point-to-Point Protocol Utility

Explanation: A break character was not transmitted to the specified device for the reason given in the accompanying message.

User Action: Take action based on the accompanying message.

CALLBACKERR, error calling network callback

Facility: PPPD, Point-to-Point Protocol Utility

Explanation: While negotiating the initial point-to-point connection, the network callback routine returned with a bad status block for the reason described in the accompanying message.

User Action: Take action based on the accompanying message.

CBLOCKMAPTOOBIG, compression block map too big for SDA to handle; may not be possible to copy dump correctly

Facility: SDA, System Dump Analyzer

Explanation: The compression block map in the dump is too big for SDA to read. In override mode, attempts to copy the dump fail if the portion of the compression block map actually used is too big for SDA to read. In normal operation, SDA attempts to read the entire compression block map, and fails only if it is unable to allocate sufficient memory for the map.

User Action: None.

CHKDOM, ExtAuth set for 'account'; ensure corresponding domain account exists

Facility: UAF, Authorize Utility

Explanation: The specified account uses external authentication and must have a corresponding account in the external authentication domain.

User Action: Verify that an account exists in the external authentication domain and that it maps to the specified OpenVMS account.

CLOSERR, unable to close file 'file-name'

Facility: SDP, Set Dump Priority Utility

Explanation: An error occurred while trying to close the specified data file.

User Action: Take action based on the accompanying message. If that message does not provide enough information to correct the error, contact a Digital support representative.

CMDNOTALLINPIP, the specified command sequence is not allowed in a PIPE command

Facility: CLI, Command Language Interpreter (DCL)

Explanation: You specified a command sequence that is illegal in a PIPE command. Refer to DCL Help or the *OpenVMS DCL Dictionary* for details.

User Action: Remove the offending command sequence from the PIPE command.

CMDPROC, error processing command

Facility: PPPD, Point-to-Point Protocol Utility

Explanation: An error occurred while processing the current PPPD command for the reason described in the accompanying message.

User Action: Take action based on the accompanying message.

CNFLTQUALS, conflicting qualifiers

Facility: SDA, System Dump Analyzer

Explanation: The SDA command contains conflicting qualifiers.

User Action: Correct or remove the conflicting qualifiers and reissue the command.

CONFIRMINPUT, SYS$COMMAND is not a terminal; /CONFIRM ignored

Facility: SYSMAN, System Management Utility

Explanation: A SYSMAN command with the /CONFIRM qualifier was submitted from something other than a terminal device.

User Action: Either execute the SYSMAN command from a terminal device, or eliminate the /CONFIRM qualifier from the SYSMAN command.

CONNECTERR, unable to connect to file 'file-name'

Facility: SDP, Set Dump Priority Utility

Explanation: An error occurred while trying to connect to the specified data file.

User Action: Take action based on the accompanying message. If that message does not provide enough information to correct the error, contact a Digital support representative.

CONNECTTERM, converting connection on device 'device-name' to a Point-to-Point connection

Facility: PPPD, Point-to-Point Protocol Utility

Explanation: PPPD has successfully accessed the specified device and is switching the serial line into Point-to-Point Protocol (PPP) mode.

User Action: None.

CREATED_SHPT, section did not exist; has been created with shared page tables

Facility: SYSTEM, System Services

Explanation: The named global section was successfully created with shared page tables.

User Action: None.

DATAOVERUN, data overrun on device 'device-name'

Facility: PPPD, Point-to-Point Protocol Utility

Explanation: The transmitting device is sending data too quickly. The specified receiving device cannot handle the volume of incoming data and has started to drop characters.

User Action: Do one of the following:

1. Set the flow control on the transmitting device to band control using the PPPD command SET/FLOW_CONTROL=XON_XOFF [ddcu:].
2. If the problem persists, exit the utility and have someone with PHY_IO privileges (usually the system manager) enable an alternate type-ahead buffer on the receiving device by entering the following DCL command:

```
$ SET TERMINAL/PERMANENT/ALTYPEAHD [ddcu:]
```

DIALINIT, error in DIAL_OUT initialization

Facility: PPPD, Point-to-Point Protocol Utility

Explanation: PPPD failed to initialize the device specified by the PPPD command DIAL_OUT for the reason described in the accompanying message.

User Action: Take action based on the accompanying message.

DISCONNECTERR, unable to disconnect from file 'file-name'

Facility: SDP, Set Dump Priority Utility

Explanation: An error occurred while trying to disconnect from the specified data file.

User Action: Take action based on the accompanying message. If that message does not provide enough information to correct the error, contact a Digital support representative.

DUMPMOD, dump has been modified

Facility: SDA, System Dump Analyzer

Explanation: The dump was previously accessed in override mode and has been modified. Such modifications sometimes make the dump impossible to analyze.

User Action: None. This is an informational message.

DUPREDSYN, cannot specify the same type of redirection syntax more than once in a command sequence

Facility: CLI, Command Language Interpreter (DCL)

Explanation: The same type of redirection syntax was specified more than once in a command sequence.

User Action: Remove the duplicate redirection syntax. You can specify multiple redirection syntaxes in a single command sequence, provided each syntax represents a different type of redirection.

EDITDEVICE, error editing device name 'device-name'

Facility: PPPD, Point-to-Point Protocol Utility

Explanation: An OpenVMS system service could not edit the specified device name for the reason described in the accompanying message.

User Action: Take action based on the accompanying message.

ERLBUFTOOBIG, size of error log buffers exceeds maximum for system parameter ERLBUFFERPAGES

Facility: SDA, System Dump Analyzer

Explanation: The error log buffers in the dump are too big for SDA to read. In override mode, attempts to extract the error log buffers may fail. In normal operation, SDA attempts to read all the error log buffers and fails only if it is unable to allocate sufficient memory for them.

User Action: None.

ERRMAPAUTH, error mapping or authenticating user

Facility: SYSTEM, System Services

Explanation: A general system service failure occurred while trying to map or authenticate the user.

User Action: Take action based on the accompanying message.

EXTAUTH, ExtAuth set for 'account'; field modification may have no effect

Facility: UAF, Authorize Utility

Explanation: The specified account uses external authentication, so the requested field change may have no effect on the authentication of that account. Typically, this message results from attempting to modify a password related account parameter while the ExtAuth flag is set for that account.

User Action: None.

EXTPWDERR, password could not be set by external authenticator

Facility: SET, SET PASSWORD Command

Explanation: An error occurred while the external authenticator was attempting to process the password request.

User Action: Take action based on the accompanying message.

FAILED, failed to set process dump priority

Facility: SDP, Set Dump Priority Utility

Explanation: An error occurred while the list of processes was being loaded into system memory (nonpaged pool).

User Action: Take corrective action based on the accompanying message; then retry the operation.

FILENUMCHK, file identification number check

Facility: SYSTEM, System Services

Explanation: This message can occur under either of the following conditions:

- The file header contains an invalid file number, which indicates that the disk's file structure is probably damaged and the volume index file is corrupt.
- An attempt was made to store more than 9999 files on a tape. This error might typically occur when using a large capacity tape for backups.

User Action: In the first case noted above, use the Analyze/Disk_Structure utility to repair the volume.

In the second case, you must delete a file before you can create another file on the media.

FILESEQCHK, file identification sequence number check

Facility: SYSTEM, System Services

Explanation: This message can occur under either of the following conditions:

- The file sequence number in the file header is not the expected one. The directory entry points to a file that was deleted.
- An attempt was made to store more than 9999 file sections in a file on a tape. This error might typically occur when using a large capacity tape for backups.

User Action: In the first case cited above, remove the directory entry using the following command:

```
DELETE file-spec
```

In the second case, you must delete a file section before you can add another file section to that file.

FORMAT, device name is not specified in the format ddcu

Facility: PPPD, Point-to-Point Protocol Utility

Explanation: The device name is not specified in the standard device name format (*_ddcu*).

User Action: Trace all logical names pointing to the physical device to ensure that the final definition conforms to the standard device name format (*_ddcu*).

ILLEGALACCM, illegal transmit ACCM bits are specified in value 'value'

Facility: PPPD, Point-to-Point Protocol Utility

Explanation: The PPPD command SET/TRANSMIT_ACCM [ddcu:] specified a bit value that cannot be set.

User Action: Specify a new mask value in the range of 0X0 to 0XFFFFFFFF. Check the *TCP/IP Networking on OpenVMS Systems* manual for additional restrictions on Asynchronous Control Character Map (ACCM) bit settings for PPPD.

INCREDSYN, incomplete redirection syntax; supply missing file parameter

Facility: CLI, Command Language Interpreter (DCL)

Explanation: The redirection syntax is missing the redirection file parameter.

User Action: Add the missing file parameter to the redirection syntax.

INUSE, cannot connect; device is in use

Facility: PPPD, Point-to-Point Protocol Utility

Explanation: An attempt was made to connect to a device that is in use.

User Action: Retry the connection when the device becomes available.

INVALID_HW_ID, invalid hardware identifier

Facility: LICENSE, License Management Utility

Explanation: The hardware identifier required by this licensed product is not on the available licenses.

User Action: Contact the software vendor for the correct license.

INVALID_TOKEN, invalid product token

Facility: LICENSE, License Management Utility

Explanation: The product token required by this licensed product is not on the available licenses.

User Action: Contact the software vendor for the correct license.

INVASNCODE, invalid ASN item code of 'value'

Facility: PPPD, Point-to-Point Protocol Utility

Explanation: The specified ASN item code is not valid.

User Action: Contact a Digital support representative.

INVCONCHAR, invalid continuation character; a command sequence cannot be terminated by a continuation character

Facility: CLI, Command Language Interpreter (DCL)

Explanation: A DCL continuation character was detected in a DCL PIPE command sequence, where it is meaningless. The command line continuation function is legal only in the context of the whole PIPE command.

User Action: Remove the continuation character that terminates the command sequence.

INVDIALSTS, invalid DIALOUT_MAIN return status of 'value'

Facility: PPPD, Point-to-Point Protocol Utility

Explanation: The specified I/O status returned by the PPPD command DIAL_OUT [ddcu:] is not valid.

User Action: Contact a Digital support representative.

INVEUSRLEN, invalid external username length

Facility: SET, SET PASSWORD Command

Explanation: The external user name entered at the prompt exceeds the maximum number of characters allowed.

User Action: Enter a user name that does not exceed the maximum number of characters permitted by the authentication modules on your system.

INVFLOWCTRL, cannot set hardware flow control on a serial port set to both NOMODEM and NOCOMMSYNC

Facility: PPPD, Point-to-Point Protocol Utility

Explanation: The flow control setting for the current serial port is not valid.

User Action: Use the XON_XOFF flow control setting or have your system manager reset the communications port to MODEM and COMMSYNC.

INVHANGASTINIT, invalid P1 passed in, P1 and io_port_block = 'value1' 'value2'

Facility: PPPD, Point-to-Point Protocol Utility

Explanation: An unexpected parameter was passed to the hangup AST during the current DIAL_OUT operation.

User Action: Contact a Digital support representative.

INVHANGASTP1, unexpected condition in hangup AST, P1 is default

Facility: PPPD, Point-to-Point Protocol Utility

Explanation: The parameter that was passed to the hangup AST was not set to SS$_HANGUP as expected during the current DIAL_OUT operation.

User Action: Contact a Digital support representative.

INVNULLCMD, null command sequence found; please specify the missing command sequence

Facility: CLI, Command Language Interpreter (DCL)

Explanation: An expected command sequence was not found.

User Action: Either add the missing command sequence or remove the extraneous command sequence separators.

INVPERMCONN, cannot establish a permanent point-to-point connection with an incoming login session

Facility: PPPD, Point-to-Point Protocol Utility

Explanation: An invalid connection was attempted through the current login session. PPPD supports only temporary (transient) connections through incoming sessions.

User Action: Establish a transient session using the PPPD command CONNECT [ddcu:].

INVPERMFLOW, cannot open a permanent point-to-point connection on a serial port set to MODEM or COMMSYNC

Facility: PPPD, Point-to-Point Protocol Utility

Explanation: An invalid connection was attempted through the current serial port. PPPD supports only temporary (transient) connections through a serial port set to MODEM or COMMSYNC.

User Action: Establish a transient session using the PPPD command CONNECT [ddcu:] or have your system manager reset the communications port to NOMODEM and NOCOMMSYNC.

INVPPPCODE, invalid PPPD item code of 'value'

Facility: PPPD, Point-to-Point Protocol Utility

Explanation: The specified PPPD item code is not valid.

User Action: Contact a Digital support representative.

INVRANGE, value specified for '/qualifier' ('value') is not within the valid range

Facility: PPPD, Point-to-Point Protocol Utility

Explanation: A qualifier specified a value that is outside the legal range.

User Action: Check the *TCP/IP Networking on OpenVMS Systems* manual for the valid values for this PPPD qualifier. Specify a value within the legal range.

INVREDSYN, invalid redirection type specified in a pipe segment

Facility: CLI, Command Language Interpreter (DCL)

Explanation: The redirection type specified in the pipe segment command in a pipeline is invalid.

User Action: Observe the following restrictions and correct the invalid redirection syntax:

- SYS$OUTPUT redirection can be specified only in the last pipe segment command of a pipeline.
- SYS$INPUT redirection can be specified only in the first pipe segment command of a pipeline.

INVUIC, invalid UIC

Facility: SYSMAN, System Management Utility

Explanation: An invalid UIC was entered. Typical errors include the following:

- The UIC format is incorrect.
- An alphanumeric UIC does not have a numeric equivalent.
- The user-supplied value for the /GROUP qualifier is not a valid UIC group number.

User Action: Enter a valid UIC. (See the *OpenVMS System Manager's Manual* for more information about UICs.) If you use the /GROUP qualifier, specify a value between 0 and 255.

INVUSER, no such user

Facility: SYSTEM, System Services

Explanation: In an attempt to map a user ID, no valid user name match was found.

User Action: If the user ID for an ExtAuth user exists in the external authentication domain, ensure that it maps to a valid OpenVMS account in SYSUAF.DAT and that the ExtAuth flag is set for the user. For locally authenticated users, ensure that the user name exists in SYSUAF.DAT.

INVVALUE, value specified for '/qualifier' ('value') is not valid

Facility: PPPD, Point-to-Point Protocol Utility

Explanation: The value given for the specified qualifier is not valid.

User Action: Check the *TCP/IP Networking on OpenVMS Systems* manual for the valid value(s) for this PPPD qualifier. Specify a legal value.

IOSBALLOC, allocated IOSB list is not empty

Facility: PPPD, Point-to-Point Protocol Utility

Explanation: When you tried to switch or disconnect the line during a DIAL_OUT operation, PPPD encountered I/O status blocks (IOSBs) from a pending I/O request in the IOSB allocation queue.

User Action: Contact a Digital support representative.

IVGROUP, invalid UIC group: 'number'

Facility: SDP, Set Dump Priority Utility

Explanation: The specified UIC group number is invalid.

User Action: Edit the data file to specify an octal group number between 1 and 37777; then retry the operation.

IVIDENT, invalid identifier: 'identifier'

Facility: SDP, Set Dump Priority Utility

Explanation: The specified identifier cannot be translated to a UIC. For example, the identifier does not exist or the RIGHTSLIST.DAT file is not accessible.

User Action: Edit the data file or otherwise correct the problem; then retry the operation.

IVMEMBER, invalid UIC member: 'number'

Facility: SDP, Set Dump Priority Utility

Explanation: The specified UIC member number is invalid.

User Action: Edit the data file to supply an octal member number between 1 to 177777; then retry the operation.

IVPROCESS, process name too long: 'process-name'

Facility: SDP, Set Dump Priority Utility

Explanation: The specified process name exceeds 15 characters.

User Action: Edit the data file to correct the problem; then retry the operation.

LBLNOTALLINPIP, labels are not allowed in a PIPE command

Facility: CLI, Command Language Interpreter (DCL)

Explanation: A DCL label was specified in a PIPE command.

User Action: Remove the offending DCL labels from the command.

LEN_NOTPAGMULT, specified length is not a multiple of CPU-specific pages

Facility: SYSTEM, System Services

Explanation: The specified length is not an even multiple of CPU-specific pages. Or, if the specified range of virtual addresses is within a shared page table region, the specified length is not an even multiple of CPU-specific page table pages or does not include the last page in the global section.

User Action: Ensure that the length passed to the system service is an even multiple of CPU-specific pages. Or, if the specified range of virtual addresses is within a shared page table region, ensure that the specified length is an even multiple of CPU-specific page table pages or includes the last page in the global section.

LENPWDSYNCH, password could not be synchronized; illegal password length

Facility: SET, SET PASSWORD Command

Explanation: A password maintained by an external authenticator cannot be set in the UAF because the password length is illegal. The password is sent to the external authenticator, but no attempt is made to synchronize the password in the UAF.

User Action: None.

LINKTIMEMISM, link time of 'file-name' ('dd-mmm-yy hh:mm') does not match link time of image in 'target' ('dd-mmm-yy hh:mm')

Facility: SDA, System Dump Analyzer

Explanation: The copy of SYS$BASE_IMAGE.EXE used by SDA to build its symbol table is not the same version as that used when the system being analyzed was booted.

This message is only a warning. SDA continues unless the mismatch causes an unrecoverable error.

User Action: Ensure that logical name ALPHA$LOADABLE_IMAGES points to a directory containing the same version of SYS$BASE_IMAGE.EXE that was used when the system being analyzed was booted.

LISTIGNORED, combination set for 'producer' 'product' includes a PAK without an include or exclude list; other lists ignored

Facility: LICENSE, License Management Utility

Explanation: During the process of combining licenses prior to loading, the License Management Facility (LMF) detected a PAK in the set without an include list or exclude list. Under these conditions, LMF essentially ignores the include lists and/or exclude lists on the other licenses and always attempts to load the license.

User Action: Issue the LICENSE MODIFY command to attach an appropriate include list or exclude list on the PAK that is missing it.

LOGNAME, logical name exceeds maximum length for device name

Facility: PPPD, Point-to-Point Protocol Utility

Explanation: The specified logical name exceeds the limit for a device name.

User Action: Revise the logical name so it does not exceed 64 characters.

MEMNOTACC, memory locations not accessible

Facility: SDA, System Dump Analyzer

Explanation: The dump is being examined in override mode. In this mode, no commands that access the dump using a memory address can be used (for example, the SHOW SUMMARY command).

User Action: In override mode, use only commands such as SET OUTPUT and SHOW DUMP.

MISSINGQUALS, needed qualifiers not given

Facility: SDA, System Dump Analyzer

Explanation: Some required qualifiers were omitted from the command.

User Action: Review the documentation and reissue the command with all the required qualifiers.

MODEMHANGUP, modem has signaled a hangup on the line; session ended

Facility: PPPD, Point-to-Point Protocol Utility

Explanation: Due to a hardware error, interference over the serial line, or a timeout event, the modem has initiated a hangup and ended data transmission over the serial line.

User Action: Try to access a better connection by repeating the previous DIAL_OUT operation. If the problem persists, check your serial cable connections, and verify that both the transmitting and receiving devices are functioning properly.

If the modem continues to initiate a hangup, notify your system manager.

MRES_PFNSMALL, specified length is larger than reserved pre-allocated memory

Facility: SYSTEM, System Services

Explanation: The amount of preallocated memory in the Reserved Memory Registry is smaller than the length specified for the global section.

User Action: Specify a smaller size for the global section, or free the preallocated memory from the Reserved Memory Registry and attempt to create the global section with the same length.

NEEDVIRTTERM, point-to-point connections on 'device-name' must be done on a virtual terminal

Facility: PPPD, Point-to-Point Protocol Utility

Explanation: The specified device does not have an associated virtual terminal session enabled.

User Action: Contact your system manager and request that they do one of the following:

- Enable a virtual terminal session on the device that you originally specified

- Designate another device on which a virtual terminal session is currently enabled

Reissue the PPPD command CONNECT [ddcu:] and specify the appropriate device.

NOASNACCESS, error accessing ASN device

Facility: PPPD, Point-to-Point Protocol Utility

Explanation: PPPD failed to access the specified asynchronous device for the reason described in the accompanying message.

User Action: Take action based on the accompanying message.

NOCHANGE, value not changed; dump not updated

Facility: SDA, System Dump Analyzer

Explanation: No change resulted from executing a MODIFY DUMP command because the location already contains the value specified in the command.

User Action: Check the location and value specified in the MODIFY command against the contents of the location to be modified; then correct the command and reissue it.

NOCLOSEBRACKET, no ']' found: 'text'

Facility: SDP, Set Dump Priority Utility

Explanation: A right bracket is missing from a process description.

User Action: Specify the process using one of the following formats:

```
[ident]process
[group,member]process
```

NODELPERMANENT, permanent persona cannot be deleted

Facility: SYSTEM, System Services

Explanation: The $PERSONA_DELETE system service detected an attempt to delete a permanent persona. Currently the only permanent persona is the process's login persona (persona ID 1).

User Action: Modify the user code that performs the $PERSONA_DELETE system service so it does not attempt to delete the permanent persona.

NODISCPRIV, no privilege to disconnect the specified device

Facility: PPPD, Point-to-Point Protocol Utility

Explanation: You do not have the privileges required to disconnect the device. Unless you have SYSPRV privileges, you cannot disconnect a point-to-point connection that you did not make.

User Action: Ask a user with SYSPRV privileges (usually the system manager) to disconnect the device.

NODUMPHERE, only the error log buffers were written to this file

Facility: SDA, System Dump Analyzer

Explanation: The ANALYZE/CRASH command was run on the stub file, SYS$ERRLOG.DMP. This file looks like a dump file, but contains only the header and error log buffers.

User Action: Analyze the correct file, either SYSDUMP.DMP or PAGEFILE.SYS.

NOEXTAUTH, external authentication is not available

Facility: SYSTEM, System Services

Explanation: External authentication is not available for authentication of the specified user. Typically, this is caused by external authentication not being enabled.

User Action: Ensure that external authentication is enabled through the SYS$SINGLE_SIGNON system logical.

NOMATCH, expected value does not match value in dump; dump not updated

Facility: SDA, System Dump Analyzer

Explanation: A MODIFY DUMP command was issued with the /CONFIRM qualifier, but the contents of the specified location in the dump do not match the value specified by /CONFIRM.

User Action: Check the location and value specified in the MODIFY command against the contents of the location to be modified; then correct the command and reissue it.

NOMEMRESID, requires rights identifier VMS$MEM_RESIDENT_USER

Facility: SYSTEM, System Services

Explanation: The program attempted to create a memory-resident global section without having the VMS$MEM_RESIDENT_USER identifier required for this privileged operation.

User Action: Enable the VMS$MEM_RESIDENT_USER identifier for the process. If you have been granted the VMS$MEM_RESIDENT_USER identifier, you can enable the identifier by issuing the following DCL command:

```
$ SET RIGHTS_LIST /ENABLE VMS$MEM_RESIDENT_USER
```

This DCL command fails if you have not been granted the VMS$MEM_RESIDENT_USER identifier. If your system manager chooses to grant you this identifier, he or she can do so by entering the following AUTHORIZE command:

```
UAF> GRANT/ID/ATTR=DYNAMIC VMS$MEM_RESIDENT_USER username
```

Once the system manager grants you the identifier, you must log out and then log back in for your process to have this identifier. For more information about Authorize utility commands, see the *OpenVMS System Management Utilities Reference Manual*.

NOMODIFY, conflicting access or protection of dump file prevents modification

Facility: SDA, System Dump Analyzer

Explanation: SDA could not open the dump file for write access when invoked with the command ANALYZE/CRASH_DUMP/OVERRIDE. Typical causes of this failure are file protection preventing write access, or another process or the system already having the file open for write access.

User Action: The user action depends upon the cause of the failure:

- If the problem is caused by file protection, use an account that has write access to the file.

- If another process already has the file open for write access, end access to the file from that process for write access, end access to the file from that process (probably by exiting SDA). Then restart SDA so that it can open the file for writing.
- If the file is held open by the system (as is the case for SYS$SYSTEM:SYSDUMP.DMP), you must make a copy of the dump file and modify the copy. You can use the DCL command COPY if you cannot use SDA because of the problem with the dump file that necessitated using the MODIFY DUMP command.

NOOPENBRACKET, no '[' found: 'text'

Facility: SDP, Set Dump Priority Utility

Explanation: A left bracket is missing from a process description.

User Action: Specify the process using one of the following formats:

```
[ident]process
[group,member]process
```

NOPARITY, device must be set to no parity

Facility: PPPD, Point-to-Point Protocol Utility

Explanation: The physical device is set to something other than no parity.

User Action: Set the physical device to no parity by executing the following command:

```
$ SET TERMINAL/NOPARITY device-name
```

NOPOOLTYPE, no pool type specified for given subtype

Facility: SDA, System Dump Analyzer

Explanation: A SHOW POOL command containing the qualifier /SUBTYPE=*n* must also include a /TYPE=*type* qualifier specifying a numeric or symbolic pool type.

User Action: Reissue the command, including the appropriate /TYPE qualifier.

NOPORTACCESS, error accessing 'device-name'

Facility: PPPD, Point-to-Point Protocol Utility

Explanation: PPPD failed to access the specified device for the reason described in the accompanying message.

User Action: Take action based on the accompanying message.

NOPPP, network protocol has not registered a PPPD callback routine

Facility: PPPD, Point-to-Point Protocol Utility

Explanation: The network protocol that you are using does not include Point-to-Point Protocol support.

User Action: Check the manufacturer's specifications and use a network protocol that includes Point-to-Point Protocol support.

NOPPPACCESS, error accessing PPP device

Facility: PPPD, Point-to-Point Protocol Utility

Explanation: PPPD failed to access the specified Point-to-Point Protocol device for the reason given in the accompanying message.

User Action: Take action based on the accompanying message.

NOPROCESSES, no processes set

Facility: SDP, Set Dump Priority Utility

Explanation: No valid process names are specified in the data file.

User Action: No action is required if you intentionally specified no process names in order to clear a previously defined process list. If the empty list is an error, edit the data file to correct the problem; then retry the operation.

NOPWDSYNCH, password could not be set; UAF synchronization is disabled

Facility: SET, SET PASSWORD Command

Explanation: The remote user ID cannot be determined and password synchronization is disabled. The SYSUAF is not modified if the password is maintained by an external authenticator.

User Action: None.

NORMRONVAX, no reserved memory registry on VAX nodes

Facility: SMI, System Management Integrator/Server

Explanation: The user issued a RESERVED_MEMORY ADD, REMOVE, or LIST command in a SYSMAN environment that includes VAX nodes.

User Action: To avoid this message, set the SYSMAN environment to include only Alpha nodes.

NOSELWGHTMOD, selection weight cannot be modified on an Availability PAK, 'product'

Facility: LICENSE, License Management Utility

Explanation: You tried to modify the selection weight of an Availability PAK.

User Action: See the *OpenVMS License Management Utility Manual* for details about using selection weights.

NOSHPTS, service cannot accept a shared page table region

Facility: SYSTEM, System Services

Explanation: The specified system service does not operate on virtual addresses within shared page table regions.

User Action: Do not specify a shared page table region when calling this system service.

NOSUCHPOOL, no 'pool-type' pool found; look in non-paged pool

Facility: SDA, System Dump Analyzer

Explanation: No pool of the specified type has been allocated. All requests for the specified type of pool have been satisfied from nonpaged pool.

User Action: None.

NOTALPHADUMP, dump file does not contain an OpenVMS Alpha system dump

Facility: SDA, System Dump Analyzer

Explanation: The dump file you are trying to analyze using the OpenVMS Alpha System Dump Analyzer was not written on an OpenVMS Alpha system.

User Action: Confirm that the dump file you wish to analyze is from an OpenVMS Alpha system, and that a system dump has actually been written to the file.

NOTANLMB, specified block is not an LMB header

Facility: SDA, System Dump Analyzer

Explanation: A SHOW DUMP/LMB=*n* command was issued where the block specified by "*n*" in the dump is not a logical memory block (LMB) header.

User Action: Review the contents of the dump and the specified SHOW DUMP command; then reissue the command, specifying a valid LMB header block number.

NOTCONNECT, device has not been set up for a point-to-point connection

Facility: PPPD, Point-to-Point Protocol Utility

Explanation: You tried to disconnect a device that was not part of a Point-to-Point Protocol (PPP) connection.

User Action: Check the device name and reissue the PPPD command DISCONNECT [ddcu:].

NOTNUMERIC, value specified for '/qualifier' ('value') must be numeric

Facility: PPPD, Point-to-Point Protocol Utility

Explanation: A nonnumeric value was entered where a numeric value is required.

User Action: Enter a numeric value.

NOTREG, network protocol has not been registered

Facility: PPPD, Point-to-Point Protocol Utility

Explanation: The protocol specified by the PPPD command SET /NETWORK_PROTOCOL [ddcu:] does not match the name of the protocol registered in the network database (using the DCL command SET NETWORK/NETWORK_TYPE).

User Action: Verify that the protocol name is correct and reissue the command. If the problem persists, contact your system manager.

NULLPIPE, null pipe segment found; please specify the missing pipe segment

Facility: CLI, Command Language Interpreter (DCL)

Explanation: An expected pipe segment command was not found in a pipeline.

User Action: Either add the missing pipe segment commands or remove the extraneous pipe segment command separator (|).

OFF_NOTPAGALGN, specified offset is not CPU-specific page aligned

Facility: SYSTEM, System Services

Explanation: The specified page offset is not CPU-specific page aligned. Or, if the specified virtual address range is within a shared page table region, the specified offset is not CPU-specific page table page aligned.

User Action: Ensure that the offset passed to the system service is CPU-specific page aligned. Or, if the specified virtual address range is within a shared page table region, ensure that the offset is CPU-specific page table page aligned.

OPENERR, unable to open file 'file-name'

Facility: SDP, Set Dump Priority Utility

Explanation: The specified data file is not accessible for the reason specified in an accompanying message.

User Action: Take action based on the accompanying message; then retry the operation.

OUTPUTERR, error creating output line

Facility: PPPD, Point-to-Point Protocol Utility

Explanation: PPPD cannot display messages or show communications settings for the reason given in the accompanying message.

User Action: Take action based on the accompanying message.

PAUSEINPUT, SYS$COMMAND is not a terminal; /PAUSE ignored

Facility: SYSMAN, System Management Utility

Explanation: A SYSMAN command with the /PAUSE qualifier was submitted from something other than a terminal device.

User Action: Either execute the SYSMAN command from a terminal device, or eliminate the /PAUSE qualifier from the SYSMAN command.

PORTDISC, error disconnecting 'device-name'

Facility: PPPD, Point-to-Point Protocol Utility

Explanation: PPPD cannot disconnect the specified device for the reason given in the accompanying message.

User Action: Take action based on the accompanying message.

PORTSETUP, error with 'device-name' device setup

Facility: PPPD, Point-to-Point Protocol Utility

Explanation: PPPD cannot configure the serial port associated with the specified device for the reason given in the accompanying message.

User Action: Take action based on the accompanying message.

PPPCONNECTERR, error connecting to PPP device

Facility: PPPD, Point-to-Point Protocol Utility

Explanation: A point-to-point connection to the specified device cannot be established for the reason given in the accompanying message.

User Action: Take action based on the accompanying message.

PPPCREATEERR, error creating link between PPP device and serial device

Facility: PPPD, Point-to-Point Protocol Utility

Explanation: A point-to-point connection between the specified devices cannot be established for the reason given in the accompanying message.

User Action: Take action based on the accompanying message.

PPPENABLEERR, error starting PPP connection

Facility: PPPD, Point-to-Point Protocol Utility

Explanation: A point-to-point connection was not enabled for the reason given in the accompanying message.

User Action: Take action based on the accompanying message.

PPPINFOERR, error obtaining information from PPP device

Facility: PPPD, Point-to-Point Protocol Utility

Explanation: PPPD cannot retrieve information from the specified device for the reason given in the accompanying message.

User Action: Take action based on the accompanying message.

PPPLISTOVER, PPP itemlist overrun; tag, items size, items item_length and len = 'value1' 'value2' 'value3' 'value4'

Facility: PPPD, Point-to-Point Protocol Utility

Explanation: A data encoding error occurred within PPPD.

User Action: Contact a Digital support representative.

PPPNOTAVAIL, point-to-point driver is not installed

Facility: PPPD, Point-to-Point Protocol Utility

Explanation: PPPD cannot locate the Point-to-Point Protocol (PPP) device driver; it is not installed on the system.

User Action: Contact your system manager.

PPPSETERR, error setting PPP device

Facility: PPPD, Point-to-Point Protocol Utility

Explanation: While the Point-to-Point Protocol (PPP) device was being initialized, an error occurred for the reason given in the accompanying message.

User Action: Take action based on the accompanying message.

PREVSET, device has already been set up for point-to-point connections

Facility: PPPD, Point-to-Point Protocol Utility

Explanation: The device specified by the PPPD CONNECT command has already been set up for point-to-point connections by a previous PPPD CONNECT command.

User Action: To reinstate a failed connection on a device, execute a PPPD DISCONNECT [ddcu:] command followed by a PPPD CONNECT [ddcu:] command.

PROCNOTDUMPED, process 'process-name' not dumped; process private memory not accessible

Facility: SDA, System Dump Analyzer

Explanation: In a selective memory dump, there is not always enough space to dump all processes. The specified process was not dumped, so only structures in system space (such as the process control block and process header) are accessible.

User Action: If processes you need to analyze are sometimes not included or are incompletely included in system dumps, consider increasing the size of the system dump file and/or use the SYS$SET_DUMP_PRIORITY utility to force the process to be written earlier in the dump.

PROCPARTDUMPED, process 'process-name' not completely dumped; some process private memory not accessible

Facility: SDA, System Dump Analyzer

Explanation: In a selective memory dump, there is not always enough space to dump all processes. The specified process was only partly dumped, so only some locations in the process's address space are accessible (in addition to structures in system space, such as the process control block and process header).

User Action: If processes you need to analyze are sometimes not included or are incompletely included in system dumps, consider increasing the size of the system dump file and/or use the SYS$SET_DUMP_PRIORITY utility to force the process to be written earlier in the dump.

PROGINIT, error in initialization

Facility: PPPD, Point-to-Point Protocol Utility

Explanation: PPPD cannot initialize itself for the reason given in the accompanying message.

User Action: Take action based on the accompanying message.

PROTOERR, error initiating network protocol callback routine

Facility: PPPD, Point-to-Point Protocol Utility

Explanation: An error occurred while attempting to execute the network protocol callback routine.

User Action: Check the manufacturer's specification to see if the network protocol supports Point-to-Point Protocol. You must use a network protocol that includes this support. If you get this message while using a network protocol that supports Point-to-Point Protocol, contact the manufacturer.

READERR, error reading from 'device-name'

Facility: PPPD, Point-to-Point Protocol Utility

Explanation: During a read operation on the specified device, an error occurred for the reason described in the accompanying message.

User Action: Take action based on the accompanying message.

READERR, unable to read from file 'file-name'

Facility: SDP, Set Dump Priority Utility

Explanation: A record in the specified data file is unreadable for the reason given in an accompanying message (for example, the record is too long).

User Action: Edit the data file to correct the problem; then retry the operation.

REMOTE, physical device specified cannot be a remote terminal

Facility: PPPD, Point-to-Point Protocol Utility

Explanation: The specified physical device is a remote terminal.

User Action: Specify a physical device that is not a remote terminal.

RMRDUPINV, name and group already in registry

Facility: SMI, System Management Integrator/Server

Explanation: The user issued a RESERVED_MEMORY ADD command with values for the name argument and /GROUP qualifier that match an extant record in the current version of SYS$SYSTEM:VMS$RESERVED_MEMORY.DATA.

User Action: Specify a different name or UIC group for the new section.

RMRFILCRE, reserved memory registry file created

Facility: SMI, System Management Integrator/Server

Explanation: The user issued a RESERVED_MEMORY ADD command for the first time on this node, resulting in the creation of SYS$SYSTEM:VMS$RESERVED_MEMORY.DATA.

User Action: None.

RMRFREPAG, pages successfully freed from reservation

Facility: SMI, System Management Integrator/Server

Explanation: The user issued a RESERVED_MEMORY FREE command that completed successfully.

User Action: None.

RMRNOMATCH, no records matched search criteria

Facility: SMI, System Management Integrator/Server

Explanation: The user issued a RESERVED_MEMORY REMOVE command with values for the name argument and /GROUP qualifier that do not match any extant records in the current version of SYS$SYSTEM:VMS$RESERVED_MEMORY.DATA.

User Action: Examine the contents of SYS$SYSTEM:VMS$RESERVED_MEMORY.DATA to confirm the name and UIC group of the record to be deleted.

RMRVERMIS, reserved memory registry version mismatch

Facility: SMI, System Management Integrator/Server

Explanation: There is a mismatch between the version marker in the current incarnation of SYS$SYSTEM:VMS$RESERVED_MEMORY.DATA and the version expected by the executing SMISERVER image.

User Action: Rename the existing version of SYS$SYSTEM:VMS$RESERVED_MEMORY.DATA and use the current version of SYSMAN to create a new registry file.

SDALINKMISM, link time of SYS$BASE_IMAGE built into SDA$SHARE ('dd-mmm-yy hh:mm') does not match link time of image in 'target' ('dd-mmm-yy hh:mm')

Facility: SDA, System Dump Analyzer

Explanation: The copy of the SDA$SHARE image you are using was not built for the version of SYS$BASE_IMAGE.EXE that was used when the system being analyzed was booted.

This message is only a warning. SDA continues unless the mismatch causes an unrecoverable error.

User Action: Be sure to use the SDA$SHARE image that was built for the OpenVMS version used in the system or dump you are analyzing.

SHORTDUMP, dump file was 'number' blocks too small when dump written; analysis may not be possible

Facility: SDA, System Dump Analyzer

Explanation: On OpenVMS Alpha systems, this message indicates that the system dump file is not large enough to accommodate either of the following:

- All physical memory (in the case of a full dump)
- Page table space, system space (S0/S1/S2), and the current process (in the case of a selective dump)

To complete the dump, you need additional blocks, as specified by 'number' in the message.

User Action: Refer to the *OpenVMS System Manager's Manual* for information about changing the size of the system dump file.

SHORTERLDUMP, error log dump file too small for error log buffers

Facility: SDA, System Dump Analyzer

Explanation: The file SYS$SYSTEM:SYS$ERRLOG.DMP is too small to hold all the error log buffers in the system.

User Action: Use AUTOGEN to ensure the file is sized correctly for your system.

SNDEXTAUTH, sending password request to external authenticator

Facility: SET, SET PASSWORD Command

Explanation: The SET PASSWORD command is being sent to the external authenticator.

User Action: None.

SUCCESS, process dump priority successfully set

Facility: SDP, Set Dump Priority Utility

Explanation: The Set Dump Priority utility successfully defined a list of processes.

User Action: None.

SYNPWDSYNCH, password could not be synchronized; illegal password syntax

Facility: SET, SET PASSWORD Command

Explanation: A password maintained by an external authenticator cannot be set in the UAF because the password syntax is illegal. The password is sent to the external authenticator, but no attempt is made to synchronize the password in the UAF.

User Action: None.

TERMINAL, physical device specified must be a terminal device

Facility: PPPD, Point-to-Point Protocol Utility

Explanation: The specified physical device is not a terminal device.

User Action: Specify a physical device that is a terminal device.

TERMSETUP, error with terminal setup

Facility: PPPD, Point-to-Point Protocol Utility

Explanation: PPPD cannot configure the host terminal for the reason given in the accompanying message.

User Action: Take action based on the accompanying message.

TERMUNSUPP, point-to-point connections on `'device-name'` are not supported

Facility: PPPD, Point-to-Point Protocol Utility

Explanation: The device to which you tried to establish a Point-to-Point Protocol (PPP) connection is not supported by PPPD.

User Action: Check the *TCP/IP Networking on OpenVMS Systems* manual to see a list of the current device restrictions for PPPD. Reenter a valid device name.

TIMERSETUP, error setting up wait

Facility: PPPD, Point-to-Point Protocol Utility

Explanation: PPPD cannot set the wait state for the reason given in the accompanying message.

User Action: Take action based on the accompanying message.

TOOMANYERLBUF, number of error log buffers exceeds maximum for system parameter ERRORLOGBUFFERS

Facility: SDA, System Dump Analyzer

Explanation: There are too many error log buffers in the dump for SDA to read. In override mode, attempts to extract the error log buffers may fail. In normal operation, SDA attempts to read all the error log buffers and fails only if it is unable to allocate sufficient memory for them.

User Action: None.

TOOMANYPIPCMD, too many pipe segment commands specified

Facility: CLI, Command Language Interpreter (DCL)

Explanation: The number of pipe segment commands caused DCL to overflow its internal buffers.

User Action: Break the pipeline up into several shorter pipelines and execute them in separate PIPE commands.

TOOMUCH, 'product' cannot be loaded; too much information

Facility: LICENSE, License Management Utility

Explanation: You tried to load a license with more data than LMF can handle.

User Action: Disable the PAKs for old versions, old releases, and old termination dates that you no longer use. Remove excess and duplicate names from the license reserve list. Try again to load the license.

TRYPWDSYNCH, attempting password synchronization

Facility: SET, SET PASSWORD Command

Explanation: The external authenticator successfully performed the password operation and is now attempting to modify the password on the user's UAF record.

User Action: None.

UNTERMSUBSH, subshell syntax not delimited with matching parentheses; add missing parenthesis

Facility: CLI, Command Language Interpreter (DCL)

Explanation: A parenthesis is missing from a subshell syntax in a PIPE command.

User Action: Ensure that each subshell is enclosed by left and right parentheses.

VA_NOTPAGALGN, specified virtual address is not page aligned

Facility: SYSTEM, System Services

Explanation: The specified virtual address is not a CPU-specific page aligned address. Or, if the specified virtual address is within a shared page table region, the specified virtual address is not CPU-specific page table page aligned.

User Action: Ensure that the virtual address passed to the system service is a CPU-specific page aligned address. Or, if the specified virtual address is within a shared page table region, ensure that the virtual address is a CPU-specific page table page aligned address.

VAXDUMP, dump file FLAGS2 DMP$V_VAXDUMP should be set

Facility: SDA, System Dump Analyzer

Explanation: The dump file you are trying to analyze does not have the FLAGS2 DMP$V_VAXDUMP bit set in the dump file header. Either the dump file was created on an OpenVMS VAX system prior to Version 7.1, or another event caused the FLAGS2 DMP$V_VAXDUMP bit to be cleared.

User Action: None. The OpenVMS VAX System Dump Analyzer will continue to access this dump file.

WRGSUBSHSYN, empty subshell or subshell is not separated from other command sequences by proper separators

Facility: CLI, Command Language Interpreter (DCL)

Explanation: An offending subshell syntax is embedded in a PIPE command. Either there is a null subshell or a subshell command sequence lacks the separators necessary to distinguish it from other command sequences.

User Action: Remove or correct the offending subshell syntax.

WRITEERR, error writing to 'device-name'

Facility: PPPD, Point-to-Point Protocol Utility

Explanation: During a write operation on the specified device, an error occurred for the reason described in an accompanying message.

User Action: Take action based on the accompanying message.

Index

A

/ADDRESS=*n* qualifier, 4–24
/ALL qualifier, 3–4, 3–6, 3–9, 3–11, 4–21, 4–23, 4–24, 4–25
Arithmetic and Logical Operators, 4–32
Asynchronous transfer mode
  See ATM
ATM (Asynchronous transfer mode)
  configuring the hardware, 4–12
  using LAN emulation over, 4–8
/ATMADDRESS qualifier, 3–2, 3–5, 3–7, 3–9
ATMWORKS 350 adapter, 4–10
ATMWORKS 750 adapter, 4–10
Authentication support, 3–13

B

BACKUP$START routine, 4–2
Backup (BACKUP) utility
  new programming interface, 4–1
  /[NO]UNSHELVE qualifier, 3–11
/BAD qualifier, 4–25, 4–30
/BAP qualifier, 4–31
BIG-REQUESTS extension, 4–3
Broadcast and unknown server (BUS), 4–9
Byte and Word emulation
  instruction emulator, 4–2
  setting the breakpoint, 4–16

C

Call command, 4–17
C function prototypes, 4–2
/CHARACTERISTICS qualifier, 3–4, 3–6, 3–9, 3–11
CIPCA adapter, 3–22
  features, 3–22
Classical IP
  managing over an ATM network, 3–1
/CLIP qualifier, 3–7, 3–9
Cluster Compatibility Kit, 3–25
CLUSTER_CONFIG.COM command procedure
  Delete Root option, 3–24
  Make Root option, 3–24
CLUSTER_CONFIG_LAN.COM command procedure
  Delete Root option, 3–24
  Make Root option, 3–24
CLUSTER_CREDITS system parameter, 3–23, 3–30
/COLOR qualifier, 4–30
/COMBINE qualifier, 3–18
Configuration tables
  rebuilding on Alpha systems, 3–29
/CONFIRM qualifier, 3–28
CONVERT DEVICE_DATABASE command, 3–17
  LANCP (LAN Control Program utility), 3–17
CONVERT NODE_DATABASE command, 3–17
  LANCP (LAN Control Program utility), 3–17
Credit waits, 3–23

D

DCL commands
  PIPE command, 2–1
  PPPD command, 2–1
DE500-AA adapter, 4–4
Debug Fixup Table
  See DFT
DECamds, 3–12
DECnet Phase IV, 3–19
DECnet-Plus for OpenVMS, 3–19
  documentation, 5–1
  features, 3–19
  upgrading from DECnet Phase IV, 3–20
DECwindows X11 display server, 4–3
  BIG-REQUESTS extension, 4–3
  XC-MISC extension, 4–3
  X Logical Font description, 4–3
  XTest extension, 4–3
DEFINE DEVICE command
  /ATMADDRESS qualifier, 3–2, 3–7
  /CLIP qualifier, 3–7
  /DLL qualifier, 3–2
  /ELAN qualifier, 3–2
  /MOPDLL qualifier, 3–2
Devices
  configuring automatically, 6–1
DEVICE_NAMING system parameter, 3–30

DFT (Debug Fixup Table), 4–16
Digital TCP/IP Services for OpenVMS, 3–20
  configuring over an ELAN, 4–14
Displaying
  version
    PCSI, 3–28
/DLL qualifier, 3–2, 3–4, 3–5, 3–6
DO command
  /CONFIRM qualifier, 3–28
  /PAUSE qualifier, 3–29
Documentation changes, 5–1
DOSD (dump file off the system disk), 3–25
DOSD (Dump file off the system disk), 3–12
DSF files, 4–18
DUMP command
  /BINARY qualifier, 4–17
  /BYTE qualifier, 4–17
  /DECIMAL qualifier, 4–17
  /HEXADECIMAL qualifier, 4–17
  /LONGWORD qualifier, 4–17
  /OCTAL qualifier, 4–17
  /QUADWORD qualifier, 4–17
  /WORD qualifier, 4–17
Dump file off the system disk
  See DOSD
DUMPSTYLE system parameter, 4–33

E

ELAN configuring, 4–12
/ELAN qualifier, 3–2, 3–5, 4–27
Emulator utility, 4–2
Error log dump files, 3–12
Error messages, A–1
eSNMP software, 3–20
Ethernet media types
  selecting from a console, 4–3
EVALUATE command
  /TIME qualifier, 4–26
External authentication, 3–13
EXTRACT FILE subcommand
  in PCSI, 3–27
EXTRACT PDF subcommand
  in PCSI, 3–27
EXTRACT PTF subcommand
  in PCSI, 3–28

F

Fast Ethernet
  adapter support, 3–16, 4–4
  cabling types, 3–16
File-based autoconfiguration, 6–1
Files of software products, 3–27
Finger utility, 3–20
FORMAT command
  /PHYSICAL qualifier, 4–26
Fortran-90 debugging, 4–18
Fortran debugging
  pointers, 4–18
Freeware offering, 2–2
FTP utility
  VIEW command, 3–20

G

GBLPAGES system parameter, 3–30
GBLPAGFIL system parameter, 3–30
GIGAswitch/ATM
  configuring, 4–13

H

/HANDLER qualifier, 4–17
HTML documentation, 5–1

I

/ICOUNTERS qualifier, 4–27
IO REBUILD command
  in SYSMAN, 3–29

K

Kernel threads, 4–6
Key processes
  definition, 3–13
KZPSA adapter, 3–24
KZTSA adapter, 3–24

L

LAN Control Program (LANCP) Utility, 3–17
LANCP commands, 4–12
LAN device drivers
  SYS$EXDRIVER, 4–3
LAN emulation
  ATMWORKS 350 adapter, 4–10
  ATMWORKS 750 adapter, 4–10
  components, 4–9
  managing over an ATM network, 3–1
  supporting over an ATM network, 4–10
  topography, 4–10
  using over an ATM network, 4–8
LAN emulation client (LEC), 4–9
LAN emulation configuration server (LECS), 4–9
LAN emulation server (LES), 4–9
LAN support
  DE500-AA adapter, 4–4
LAN_FLAGS parameter, 4–11
LAN_FLAGS system parameter, 3–30
LICENSE LIST command
  /SELECTION_WEIGHT qualifier, 3–18

LICENSE LOAD command
/NOUNLOAD qualifier, 3–18
/UNLOAD qualifier, 3–18
LICENSE MODIFY command
/COMBINE qualifier, 3–18
/NOCOMBINE qualifier, 3–18
/SELECTION_WEIGHT qualifier, 3–18
LICENSE REGISTER command
/OPTIONS=USER qualifier, 3–18
/OPTIONS=VAX_ALPHA qualifier, 3–18
LINK command
/THREADS_ENABLE qualifier, 4–6
Linker utility, 4–6
LIST DEVICE command
/ALL qualifier, 3–4, 3–9
/CHARACTERISTICS qualifier, 3–4, 3–9
/DLL qualifier, 3–4
/PARAMETERS qualifier, 3–4, 3–9
Listing software
PCSI, 3–28
LIST subcommand
in PCSI, 3–28
/LOCAL_PASSWORD qualifier, 3–14
Logging in
See External authentication
LOGINOUT command
/LOCAL_PASSWORD qualifier, 3–14
/LOG qualifier, 4–26

M

MACRO–32 compiler
built-ins for byte and word emulation, 4–14
MC_SERVICES system parameters, 3–31
MEMORY CHANNEL
product overview, 3–21
Messages, A–1
Minimerge
on system disks, 3–26
MOP booting, 4–12
/MOPDLL=(option) qualifier, 3–2
MSCP_CMD_TMO system parameter, 3–31

N

Networking
supported options, 3–18
/NOCOMBINE qualifier, 3–18
/NOLOG qualifier, 4–26
/NOUNLOAD qualifier, 3–18
/[NO]UNSHELVE qualifier, 3–11
NPAG_AGGRESSIVE system parameter, 3–32
NPAG_GENTLE system parameter, 3–32
NPAG_INTERVAL system parameter, 3–32

O

OpenVMS
Freeware CD–ROM, 2–2
OpenVMS Cluster Compatibility Kit, 3–25
OpenVMS Cluster systems, 3–21
migrating satellite booting to LANCP, 3–22
mixed-architecture support, 3–25
mixed-version support, 3–25
port allocation classes, 3–24
SCSI host adapters, 3–24
warranted and migration support, 3–25
OpenVMS Debugger, 4–16
DECwindows Motif interface, 4–18
register displays, 4–18
OpenVMS Internet Product Suite, 2–2
ANU-News, 2–5
CERN Proxy Server, 2–4
DECthreads HTTP Server for OpenVMS, 2–4
DEHTML, 2–6
Digital VTX Web Interface, 2–4
ENTIFY, 2–6
fgrep, 2–5
flex, 2–5
GNU awk, 2–5
GNU bison, 2–5
GNU sed, 2–5
Gopher client, 2–4
Gopher server, 2–4
grep, 2–5
HLP2HTML Converter, 2–6
HTML2PS, 2–6
HTMLCHEK, 2–6
HTMLCHKP, 2–6
HTMLSRPL, 2–6
IUPOP3, 2–5
Lynx, 2–4
MAKEMENU, 2–6
METACHAR, 2–6
MPEG Video Player, 2–5
mx, 2–5
MXRN, 2–5
Netscape Commerce Server, 2–4
Netscape Communications Server, 2–4
Netscape Navigator, 2–4
NEWSRDR, 2–5
perl, 2–5
Pine E-mail, 2–5
Purveyor WebServer, 2–4
SDML2HTML, 2–6
Spyglass Enhanced Mosaic, 2–4
tcl, 2–5
tk, 2–5
TXT2HTML, 2–6
VNEWS, 2–5
web interface to DCL help, 2–6
web interface to system monitoring, 2–6

OpenVMS Internet Product Suite (cont'd)
WEBLINT, 2–6
xv, 2–5
yacc, 2–5
ZIP and UNZIP, 2–5
OpenVMS License Management Facility (LMF), 3–17
OpenVMS Management Station, 3–26
Operators
Arithmetic and Logical, 4–32
/OPTIONS=USER qualifier, 3–18
/OPTIONS=VAX_ALPHA qualifier, 3–18

P

/PAGE qualifier, 4–32
/PAGE_TABLES=ALL qualifier, 4–31
/PAGE_TABLES/PT qualifier, 4–31
/PARAMETERS qualifier, 3–4, 3–6, 3–9, 3–11
PARAMETERS SHOW command
/PAUSE qualifier, 3–29
Password synchronization, 3–15
PATHWORKS over ELAN
configuring, 4–14
/PAUSE qualifier, 3–29
PCSI PRODUCT command
EXTRACT FILE subcommand, 3–27
EXTRACT PDF subcommand, 3–27
EXTRACT PTF subcommand, 3–28
LIST subcommand, 3–28
SHOW UTILITY subcommand, 3–28
/PERMANENT qualifier, 3–5, 3–9
/PHYSICAL qualifier, 4–26, 4–27
PIPE command, 2–1
Point-to-Point Protocol
PPP driver, 2–7
PPPD utility, 2–7
POLYCENTER Software Installation Utility, 3–27
POP (Post Office Protocol), 3–20
Port allocation classes, 3–24
Post Office Protocol
see POP
PPPD command, 2–1
Product description file (PDF) of software products, 3–27
Product text file (PTF) of software products, 3–28

Q

/QUEUE qualifier, 4–27

R

READ command
behavior change, 4–33
/LOG qualifier, 4–26
/NOLOG qualifier, 4–26
Rebuilding configuration tables
on Alpha systems, 3–29
RMS system parameters
RMS_DFLRL parameter, 3–33
RMS_HEURISTIC parameter, 3–32
RMS_DFLRL parameter, 3–33
RMS_HEURISTIC parameter, 3–32

S

SCSI bus
fast wide differential (FWD), 3–24
SCSI devices
driver enhancements, 4–19
naming, 3–24
support for 16 devices, 3–25
SCSI host adapters
KZPSA, 3–24
KZTSA, 3–24
SCSI interconnect
multi-host, 3–24
SCSI Skip files
new function modifier, 4–19
SCSI wide adapters
support for devices, 3–16
SDA (System Dump Analyzer), 4–20
SEARCH command
/PHYSICAL qualifier, 4–27
/SELECTION_WEIGHT=n qualifier, 3–18
/SELECTION_WEIGHT qualifier, 3–18
Selective system dump files, 3–13
SET BREAK command
/HANDLER qualifier, 4–17
/SYSEMULATE qualifier, 4–16
SET DEVICE command
/ATMADDRESS qualifier, 3–5, 3–9
/CLIP qualifier, 3–9
/DLL qualifier, 3–5
/ELAN qualifier, 3–5
/PERMANENT qualifier, 3–5, 3–9
SET ERASE_SCREEN command, 4–20
SET FETCH command
/PHYSICAL qualifier, 4–27
/VIRTUAL qualifier, 4–27
SET TERMINAL command
/WRAP qualifier, 4–18
SHOW ADDRESS command, 4–20
SHOW BUGCHECK command, 4–21
SHOW DEVICE command
/ALL qualifier, 3–6, 3–11
/CHARACTERISTICS qualifier, 3–6, 3–11
/DLL qualifier, 3–6
/PARAMETERS qualifier, 3–6, 3–11
SHOW GLOBAL_SECTION_TABLE command, 4–22
SHOW GSD command, 4–23

SHOW LAN command
/ELAN qualifier, 4–27
/ICOUNTERS qualifier, 4–27
/QUEUE qualifier, 4–27
/VCI qualifier, 4–27
SHOW PAGE_TABLE command, 4–32
SHOW PFN_DATA command
/BAD qualifier, 4–30
/COLOR qualifier, 4–30
/ZERO qualifier, 4–30
SHOW POOL command
/BAP qualifier, 4–31
SHOW PROCESS command
/PAGE qualifier, 4–32
/PAGE_TABLES=ALL qualifier, 4–31
/PAGE_TABLES/PT qualifier, 4–31
SHOW RMD command, 4–24
SHOW UTILITY subcommand
in PCSI, 3–28
SHOW WORKING_SET_LIST command, 4–25
SONET/SDH framing, 4–11
SYS$ACME_MODULE logical name, 3–13
SYS$EWDRIVER LAN device driver, 4–3
SYS$SINGLE_SIGNON logical name, 3–13
/SYSEMULATE qualifier, 4–16
SYSGEN parameters
LAN_FLAGS parameter, 4–11
SYSMAN utility, 3–28
System disks
minimerge capability, 3–26
System Dump Analyzer
See SDA
System Management utility (SYSMAN)
commands
IO REBUILD, 3–29
System messages, A–1
System parameters, 3–30
System services
64-bit addressing, 4–34
C function prototypes, 4–2
new services, 4–34

T

THREADCP tool, 4–8
/THREADS_ENABLE qualifier, 4–6
/TIME qualifier, 4–26

U

/UNLOAD qualifier, 3–18

V

VALIDATE PFN_LIST command, 4–25
/VCI qualifier, 4–27
/VIRTUAL qualifier, 4–27
Volume Shadowing
minimerge function, 3–25
Volume Shadowing for OpenVMS
minimerge on system disks, 3–26

W

/WORKING_SET_LIST = GLOBAL option, 4–32
/WORKING_SET_LIST = LOCKED option, 4–32
/WORKING_SET_LIST = MODIFIED option, 4–32
/WORKING_SET_LIST = PPT option, 4–32
/WORKING_SET_LIST = PROCESS option, 4–32
/WRAP qualifier, 4–18

X

X11 R6 server, 4–3
XC-MISC extension, 4–3
X Logical Font description, 4–3
XQP+ processor, 3–20
XTest extension, 4–3

Z

/ZERO qualifier, 4–30