OpenVMS Version 6.2 Release Notes

Order Number: AA–QJEFA–TE

May 1995

This manual describes changed and enhanced features; upgrade and compatibility information; new and existing software problems and restrictions; and software and documentation corrections.

Revision/Update Information:	This is a new manual.
Software Version:	OpenVMS Version 6.2

Digital Equipment Corporation
Maynard, Massachusetts

May 1995

ZK6422

This document is available on CD–ROM.

This document was prepared using VAX DOCUMENT Version 2.1.

Send Us Your Comments

We welcome your comments on this or any other OpenVMS manual. If you have suggestions for improving a particular section or find any errors, please indicate the title, order number, chapter, section, and page number (if available). We also welcome more general comments. Your input is valuable in improving future releases of our documentation.

You can send comments to us in the following ways:

- Internet electronic mail: openvmsdoc@zko.mts.dec.com
- Fax: 603-881-0120 Attn: OpenVMS Documentation, ZKO3-4/U08
- Online form

  Print or edit the online form SYS$HELP:OPENVMSDOC_COMMENTS.TXT. Send the completed online form by electronic mail to our Internet address, or send the completed hardcopy form by fax or through the postal service.

  Please send letters or the form to:

  ```
  Digital Equipment Corporation
  OpenVMS Documentation
  110 Spit Brook Road, ZKO3-4/U08
  Nashua, NH  03062-2698
  USA
  ```

Thank you.

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: 264.3030 603.884.3030	Fax: 603.884.3960	U.S. Software Supply Business Digital Equipment Corporation 10 Cotton Road Nashua, NH 03063−1260

ZK−7654A−GE

Contents

2 General User Release Notes

3 System Management Release Notes

4 Programming Release Notes

A ISA Bus Support (Alpha Only)

B Remedial Kits Included in OpenVMS Version 6.2

C Remedial Kits Not Included in OpenVMS Version 6.2

D Layered Product Support for OpenVMS Version 6.2

Index

Examples

Figures

Tables

Preface

Intended Audience

This manual is intended for all OpenVMS operating system users. Read this manual before you install, upgrade, or use Version 6.2 of the operating system.

Document Structure

This manual contains the following chapters and appendixes:

- Chapter 1 contains release notes that pertain to installing and upgrading the OpenVMS operating systems, as well as hardware-related information.
- Chapter 2 contains release notes that relate to the general use of the OpenVMS operating system.
- Chapter 3 contains release notes specific to system management information.
- Chapter 4 contains release notes that relate to programming on an OpenVMS system.
- Appendix A contains information about Industry Standard Architecture (ISA) I/O bus support for Alpha platforms.
- Appendix B lists remedial kits that are included in OpenVMS Version 6.2.
- Appendix C describes remedial kits that are *not* included in OpenVMS Version 6.2.
- Appendix D contains tables that summarize layered product support for OpenVMS Version 6.2. The tables also identify which layered products work with DECnet/OSI synonyms and full names.

This manual contains release notes introduced in the current release and notes from previous OpenVMS versions that still apply to the new release. Margin notes for each release note indicate the version of origin (for example, V6.1).

Notes from previous releases are published when:

- The release note has not been documented in any other manual in the OpenVMS documentation set, and the note is still pertinent.
- The release note may be pertinent in multiple-version OpenVMS cluster systems.

Associated Information

For a list of additional documents that are available in support of this version of the OpenVMS operating system, refer to the *Overview of OpenVMS Documentation*.

The *Alpha Architecture Reference Manual* mentioned in this document is published by Digital Press under order number EY–L520E–DP.

Conventions

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

In this manual, every use of OpenVMS Alpha means the OpenVMS Alpha operating system, every use of OpenVMS VAX means the OpenVMS VAX operating system, and every use of OpenVMS means both the OpenVMS VAX and OpenVMS Alpha operating systems.

The following conventions are used to identify information specific to OpenVMS Alpha or OpenVMS VAX:

Alpha	The Alpha icon denotes the beginning of information specific to OpenVMS Alpha.
VAX	The VAX icon denotes the beginning of information specific to OpenVMS VAX.
♦	The diamond symbol denotes the end of a section of information specific to OpenVMS Alpha or to OpenVMS VAX.

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 press the key labeled Ctrl while you press another key or a pointing device button.
PF1 *x*	A sequence such as PF1 *x* indicates that you must first press and release the PF1 key, then press and release another key or a pointing device button.
Return	In examples, a key name enclosed in a box indicates that you press a key on the keyboard. (In text, a key name is not enclosed in a box.)
. . .	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 surround a required choice of options; you must choose one of the options listed.
boldface text	Boldface text represents the introduction of a new term or the name of an argument, an attribute, or a reason (user action that triggers a callback). Boldface text is also used to show user input in Bookreader versions of the manual.
italic text	Italic text emphasizes important information and indicates complete titles of manuals and 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.
-	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, hexadecimal—are explicitly indicated.
`struct`	Monospace type in text identifies the following C programming language elements: keywords, the names of independently compiled external functions and files, syntax summaries, and references to variables or identifiers introduced in an example.

1

OpenVMS Installation, Upgrade, and Hardware Release Notes

This chapter contains information that applies to installations and upgrades of the OpenVMS VAX and OpenVMS Alpha operating systems. It also provides information specific to certain hardware.

The installation and upgrade notes in this chapter are organized into the following categories:

- Installation and upgrade notes common to both VAX and Alpha systems (see Section 1.1)
- Alpha specific installation and upgrade notes (see Section 1.3)
- VAX specific installation and upgrade notes (see Section 1.2)

Hardware and firmware notes follow the upgrade and installation sections.

1.1 Installation and Upgrade Information Common to VAX and Alpha

The following notes document installation and upgrade information common to both platforms. For more VAX specific installation and upgrade notes, see Section 1.2. For additional Alpha specific notes, see Section 1.3.

1.1.1 Changes and Enhancements

This section describes information related to installing or upgrading the OpenVMS VAX or OpenVMS Alpha operating system.

1.1.1.1 Software Product Descriptions (SPDs)

V6.2

Printed copies of the Software Product Descriptions (SPDs) are no longer provided. However, the following SPDs are now available on the OpenVMS Version 6.2 media:

- OpenVMS Operating System for VAX and Alpha, Version 6.2
- OpenVMS Cluster Software, Version 6.2
- DECnet for OpenVMS VAX and Alpha, Version 6.2
- RMS Journaling for OpenVMS, Version 6.2
- Volume Shadowing for OpenVMS, Version 6.2

Customers who receive the operating system and documentation on CD–ROM can consult the *OpenVMS Version 6.2 CD–ROM User's Guide* for complete information about locating and printing the SPDs. VAX users who receive their distribution kit on other media should refer to the *OpenVMS VAX Version 6.2 Upgrade and Installation Manual* for information about how to locate and access the SPD files on their kit.

1.1.1.2 Using ConDIST to Update InfoServer Software

V6.1

You can use the Digital Consolidated Software Distribution (ConDIST) to update InfoServer software. After you log in to the InfoServer system, perform the following steps:

1. Insert the disc containing the [INFOSERVxxx] directory tree in a compact disc drive attached to the InfoServer system.
2. At the InfoServer> prompt, enter a command in the following format, where *n* is the drive number:
   - On the InfoServer 100 or InfoServer 150 system, enter a command in the following format:

     UPDATE SYSTEM DK*n*:
   - On the InfoServer 1000 system, enter a command in the following format:

     UPDATE SYSTEM DK*n*: FLASH

   These commands move the InfoServer software to the internal read/write device. The next time you boot the InfoServer system, it runs the updated software. Note that you can also boot the server from the ConDIST disc.

1.1.2 Problems and Restrictions

This section contains upgrade and installation problems and restrictions that are common to both VAX and Alpha operating systems.

1.1.2.1 DECnet Product Requirements

V6.2

VAX

On OpenVMS VAX systems, if you currently run DECnet/OSI, the OpenVMS VAX Version 6.2 kit detects this and gives you the option to retain those files. If you prefer to use the Phase IV code base, you have the option to do so. However, you cannot use DECnet over TCP/IP if you choose that option. ♦

Alpha

On OpenVMS Alpha systems, if DECnet/OSI is currently installed, the DECnet/OSI files are automatically retained during an upgrade. If you prefer to use DECnet Phase IV, you must do the following:

1. Ensure that the DECnet Phase IV option is selected when you upgrade to OpenVMS Alpha Version 6.2. (You can repeat the upgrade procedure to select DECnet Phase IV later.)
2. After upgrading and booting the upgrade system, execute the following command to remove DECnet/OSI:

   ```
   $ PRODUCT REMOVE DECNET_OSI ♦
   ```

If you want to run the DECnet/OSI for OpenVMS layered product on your OpenVMS Version 6.2 systems, you must first upgrade to Version 6.1 of DECnet/OSI for OpenVMS before you can install any later versions of the product. Version 6.1 of DECnet/OSI allows you to run DECnet applications over TCP/IP. Before you upgrade to Version 6.1 of DECnet/OSI for OpenVMS, note the following requirements:

If . . .	Then . . .
You are upgrading from OpenVMS Version 6.1,	You can upgrade to Version 6.1 of DECnet/OSI prior to upgrading your OpenVMS system.
You are upgrading from OpenVMS VAX Version 6.0 or OpenVMS Alpha Version 1.5,	You must upgrade your OpenVMS system before upgrading to Version 6.1 of DECnet/OSI.
You are running DEC TCP/IP Services for OpenVMS,	You must upgrade to Version 3.2 of the DEC TCP/IP Services for OpenVMS product before upgrading to Version 6.1 of DECnet/OSI.
You are running TCP/IP from another vendor,	Consult your TCP/IP vendor for the minimum version of their product that will support DECnet applications.

1.1.2.2 DECnet/OSI Installation Can Fail

V6.1

During the installation of DECnet/OSI for OpenVMS, VMSINSTAL may fail with the following error message:

```
%SMI-E-INGDUPINV, Image NET$MESSAGE for product DECNET already in images table
```

If you experience this behavior, issue the following commands and repeat the installation:

```
$ SYSMAN:=$SYSMAN
$ SYSMAN SYS_LOADABLE REMOVE DECNET NET$MESSAGE
```

Note that under normal circumstances you will not experience this behavior.

1.1.2.3 DECnet/OSI: NET$MANAGE Rights Identifier Recommended

V6.1

After installing the Decnet/OSI for OpenVMS product, Digital recommends that you grant the NET$MANAGE rights identifier to the SYSTEM account or any other account from which you expect to use VMSINSTAL. Installations of some layered products may fail if you are running Decnet/OSI and the account running the installation does not have this identifier.

1.1.2.4 InfoServer Client Can Fail to Start If DECnet Is Started or Stopped

V6.0

The InfoServer client software will fail to start on a system where DECnet has been started and subsequently stopped. The following message will be found in the file SYS$MANAGER:ESS$STARTUP.LOG:

```
%ESS-I-NONET ESS started before DECnet.  4-MAR-1994 16:36:39.29
```

This is caused by a problem in the startup procedure SYS$STARTUP:ESS$STARTUP.COM.

If the InfoServer client must be started at this point, the LASTport transport can be started with the Last Control Program using the following command:

```
$ MCR ESS$LASTCP
LASTCP> START
```

This command will start the transport. You may now execute the InfoServer client startup:

```
$ @SYS$STARTUP:ESS$STARTUP DISK
```

Because the last transport is already started, the startup will run successfully.

1.1.2.5 POLYCENTER Software Installation Utility Can Hang During Layered Product Installations

V6.2

During a layered product installation, if another process is accessing the DCL help library, the POLYCENTER Software Installation utility will hang until it can open the file for exclusive access. When all users have exited from DCL help, the utility automatically resumes installation of the layered product. This wait condition occurs after the "Portion Done: 0%..." message is displayed but before the percent complete indicator reaches 100%. However, no explicit warning message is displayed to alert the user who issued the PRODUCT INSTALL command that one or more other users have the DCL help library open.

1.1.2.6 VMSINSTAL Can Fail to Update DCL Help During Layered Product Installations

V6.0

During a layered product installation, if another process is accessing DCL Help, the following events occur:

- The installer sees the following message displayed once:

  ```
  %VMSINSTAL-I-DCLHLPINUSE, Trying to update DCL HELP library.  Procedure
  will try three more times.
  ```

  The procedure makes up to three additional attempts to access DCL Help (one attempt every 1 1/2 minutes).

- All user processes see the following message up to three times (that is, each time VMSINSTAL attempts and fails to access DCL Help):

  ```
  Software installation procedure in progress, but DCL HELP command is
  in use. Trying to update DCL HELP library.  Please exit DCL HELP
  command temporarily.
  ```

- After three tries to update DCL Help (4 1/2 minutes), if DCL Help is still accessed, VMSINSTAL does the following:

  1. Moves the files to be updated to a working directory
  2. Notifies the installer with the following message:

     ```
     %VMSINSTAL-I-NODCLHLP, DCL HELP not provided for this product.
     Manually update HELP libraries after installation.
     Use SYS$COMMON:[SYSHLP]<file name> for providing
     new HELP
     ```

  3. At the completion of the installation, issues the following message to the installer:

     ```
     %VMSINSTAL-I-REFDCLHLP, DCL HELP could not be updated.
     Reference SYS$UPDATE:DODCLHELP.VMI for information updating
     DCL HELP.
     ```

1.2 VAX Specific Installation and Upgrade Information

VAX

The release notes in this section pertain only to installations or upgrades of OpenVMS VAX operating systems. See Section 1.1 for additional notes that pertain to both VAX and Alpha systems. For complete information about installing or upgrading your OpenVMS VAX Version 6.2 operating system, refer to the *OpenVMS VAX Version 6.2 Upgrade and Installation Manual*.

1.2.1 Problems and Restrictions

This section describes problems and restrictions related to installing or upgrading OpenVMS VAX Version 6.2.

1.2.1.1 Upgrade Version Requirements

V6.2

To upgrade your system to OpenVMS VAX Version 6.2, you must have OpenVMS VAX Version 6.0 or 6.1 already installed.

To perform a concurrent upgrade or rolling upgrade in a VAXcluster environment, all nodes in the cluster must be running at least OpenVMS VAX Version 5.5–2. For rolling upgrades, Digital recommends that all nodes in the cluster run the same version of the operating system.

For more information about upgrading to OpenVMS VAX Version 6.2, see the *OpenVMS VAX Version 6.2 Upgrade and Installation Manual*.

1.2.1.2 Volume Shadowing Phase I Not Available

V6.2

Volume shadowing phase I (controller-based) is no longer available. Phase II (host-based) shadowing fully replaces phase I shadowing and provides significantly enhanced features.

For more information about migrating to phase II, see *Volume Shadowing for OpenVMS*.

At the beginning of an upgrade, the procedure checks the setting of the SHADOWING system parameter. If you have not migrated to phase II and the procedure detects the presence of phase I shadowing, it displays a message and asks you if you want to continue the upgrade.

At that point, you can exit the upgrade procedure, migrate to phase II, and restart the upgrade from the beginning. If you continue the upgrade without migrating to phase II shadowing, volume shadowing will not be present on your system after the upgrade has completed.

1.2.1.3 DECnet for OpenVMS VAX Extensions Not Supported

V6.2

The DECnet for OpenVMS VAX Extensions product is not supported for OpenVMS VAX Version 6.2. The network products associated with DECnet for OpenVMS VAX Extensions (P.S.I., WANDD, VOTS, OSAK, and FTAM) are no longer available separately. If you need these products, you must migrate to Version 6.1 of DECnet/OSI for OpenVMS. For information about planning for and implementing DECnet/OSI for OpenVMS software, see the DECnet/OSI installation and configuration manuals.

1.2.1.4 MACRO32.EXE and Standalone BACKUP

V6.2

To build standalone BACKUP onto tape, you must have the image MACRO32.EXE on your system disk. Do not tailor off the Macro libraries before building standalone BACKUP onto tape.

1.2.1.5 DECpresent Dependencies for Installing on OpenVMS VAX Version 6.1

V6.1

To run DECpresent Version 1.0A on OpenVMS VAX Version 6.1, you must upgrade the CDA Converter Library from Version 1.1 to Version 2.0.

When installing DECpresent Version 1.0A on OpenVMS VAX Version 6.1, system managers can safely ignore the IVP failure for the CDA Converter Library Version 1.1 because that version of the product is bundled with DECpresent but does not work on OpenVMS VAX Version 6.1.

After installing DECpresent Version 1.0A on OpenVMS VAX Version 6.1, or upgrading from VMS Version 5.5–2 to Version 6.1 with DECpresent Version 1.0A already installed on the system, system managers should install CDA Converter Library Version 2.0.

1.2.1.6 VAXstation and MicroVAX Installation Workaround

V6.0

This release note applies specifically to VAXstation 4000-VLC, VAXstation 4000 Model 60, VAXstation 4000 Model 90, MicroVAX 3100 Models 30 and 40, and MicroVAX 3100 Model 80.

If you halt the OpenVMS VAX Version 6.0 (or later) installation procedure after booting the new system disk, the console will either display miscellaneous characters or appear to hang. Turn the system power off and on to recover the use of the console and continue the installation.

Once AUTOGEN runs and the system reboots at the completion of the installation procedure, the console is usable again. ♦

1.3 Alpha Specific Installation and Upgrade Information

Alpha

The release notes in this section pertain only to installations or upgrades of OpenVMS Alpha operating systems. See Section 1.1 for additional notes that pertain to both Alpha and VAX systems. For complete information about installing or upgrading your OpenVMS Alpha Version 6.2 operating system, refer to the *OpenVMS Alpha Version 6.2 Upgrade and Installation Manual*.

1.3.1 Changes and Enhancements

This section describes changes and enhancements to OpenVMS Alpha installation and upgrade procedures.

1.3.1.1 Installing or Upgrading an Alpha System from a Running System

V6.2

Beginning with OpenVMS Alpha Version 6.2 systems, you can use the SYS$SYSTEM:AXPVMS$PCSI_INSTALL.COM procedure to install or upgrade from a running OpenVMS Alpha system to another target system disk. The *OpenVMS Version 6.2 New Features Manual* describes this new feature in more detail.

Using the VMSKITBLD.COM procedure to duplicate system files from an existing system disk to another disk now applies only to OpenVMS VAX systems (see the *OpenVMS System Manager's Manual: Essentials* for details).

1.3.1.2 Removing the OpenVMS Alpha Operating System

V6.1

Although use of the PRODUCT REMOVE command is not fully supported in OpenVMS Alpha Version 6.1 and later for the removal of the OpenVMS operating system, you can use the PRODUCT REMOVE command to remove most of the OpenVMS Alpha operating system from a system disk without affecting user files on the disk.

Follow these steps to remove OpenVMS Alpha:

1. If your system disk has multiple system-specific roots, boot the system and execute SYS$MANAGER:CLUSTER_CONFIG to remove all roots except the one from which you are booted.

2. Shut down and boot from the distribution CD–ROM or from a system disk other than the one from which OpenVMS Alpha is being removed. Execute the following DCL commands; substitute the device name of the disk from which OpenVMS Alpha is being removed for <target-disk>, and the root number that you did *not* remove in step 1 for SYSx.

```
$ DEFINE/NOLOG PCSI$SYSDEVICE <target-disk>
$ DEFINE/NOLOG PCSI$SPECIFIC <target-disk>:[SYSx.]
$ DEFINE/NOLOG PCSI$DESTINATION <target-disk>:[VMS$COMMON]
$ PRODUCT REMOVE VMS /REMOTE
```

   If OpenVMS Alpha is not running from the distribution CD–ROM, you will need to be logged in to a privileged account.

3. After the remove operation completes, review the target disk to determine if you want to delete the following files, which the PRODUCT REMOVE command cannot remove.

   - In <target-disk>:[SYS*.SYSEXE], where * is 0 or the hexadecimal number of any additional VMScluster roots on the target disk:

     ALPHAVMSSYS.PAR
     MODPARAMS.DAT
     PAGEFILE.SYS
     SWAPFILE.SYS

   - In <target-disk>:[VMS$COMMON.SYSEXE]:

     LMF$LICENSE.LDB
     PCSI$FILE_SYSTEM.PCSI$DATABASE
     PCSI$PROCESSOR.PCSI$DATABASE
     PCSI$ROOT.PCSI$DATABASE
     RIGHTSLIST.DAT

   Note

   Do not remove the *.PCSI$DATABASE files if you have layered products installed on this disk, or if you want to maintain a history of software installation on this disk.

4. Review the target disk for the directory structures [VMS$COMMON...] and [SYSx...], which will remain after removing OpenVMS Alpha. You may want to remove these directories.

1.3.2 Problems and Restrictions

This section describes problems and restrictions related to installing or upgrading OpenVMS Alpha Version 6.2.

1.3.2.1 RAID Devices Naming Problem

V6.2

If you have RAID devices connected to StorageWorks RAID Array 210 or 230 subsystems, you might experience device-naming problems when running in a cluster environment if nonzero allocation classes are used. In this case, the RAID devices will be named $*n*$DR*cu*, where: *n* is the (nonzero) allocation class, *c* is the controller letter, and *u* is the unit number.

If multiple nodes in the cluster have the same (nonzero) allocation class and these same nodes have RAID controllers, then RAID devices that are distinct might be given the same name (for example, 1DRA0). This problem can lead to data corruption.

A new system parameter, DR_UNIT_BASE, has been added to provide a way for unique RAID device names to be generated. The RAID controller adds DR_UNIT_BASE to the unit number, *u*, before it creates the RAID device names. With DR_UNIT_BASE set to 0, the device name 1DRA0 is produced; with the parameter set to 10, the device name 1DRA10 is produced.

Setting DR_UNIT_BASE to appropriate, nonoverlapping values on all cluster members that share the same (nonzero) allocation class ensures that no two RAID devices are given the same name.

1.3.2.2 Audio Editor in DECwindows Mail

V6.1

The Audio Editor in DECwindows Mail is not usable on systems running DECwindows Motif Version 1.1 and OpenVMS Alpha Version 6.1 or later. The following error message is displayed when you attempt to create a new mail message with the Audio Editor:

```
Application Interface library error 0
```

To eliminate this problem, install a kit for DECwindows Motif Version 1.2 or later.

1.3.2.3 Booting Messages

V6.1

You may see the following message on some systems during booting. This is simply an informational message to indicate that an image was not installed as RESIDENT due to insufficient code or data granularity hint region size.

```
%INSTALL-I-NONRES, installed image non-resident with other specified
options
-INSTALL-E-NOGHREG, insufficient memory in the code or data granularity hint
region
```

You can either ignore this message or run AUTOGEN to eliminate the error message.

For more information about the /RESIDENT qualifier, see the *OpenVMS System Management Utilities Reference Manual*. For more information on installing resident images, see the *OpenVMS Linker Utility Manual*.

1.3.3 Corrections

This section describes corrections to OpenVMS Alpha installation and upgrade procedures.

1.3.3.1 System Directories Installation

V6.2

During an installation of OpenVMS AXP Version 6.1, the following five directories were created without world read access:

```
[VMS$COMMON]
[VMS$COMMON.SYSEXE]
[SYS0]
[SYS0.SYSEXE]
[SYS0.SYSCOMMON]
```

This happened only during installation; directory protections were not changed during an upgrade.

This problem has been fixed. ♦

1.4 Alpha Firmware Revisions

Alpha

Refer to the *Alpha AXP Systems Firmware Update Version 3.2 Release Notes Overview* (Order Number AA-PW8YM-TE) contained on the Alpha Firmware Version 3.2 CD–ROM for the revision numbers of the console firmware needed for each Alpha system type.

The Release Notes Overview references a DOC directory that contains release notes documents for each system type. The release notes documents contain tables that list compatible firmware and operating system versions. ♦

1.5 AlphaStation and AlphaServer Series Computers

Alpha

This section contains release notes that apply to AlphaStation and AlphaServer systems.

1.5.1 Changes and Enhancements

This section describes graphics support on OpenVMS Alpha systems.

1.5.1.1 Console Variable Default Change on AlphaServer 2100 Systems

V6.2

Version V4.0 of the AlphaServer 2100 console now defaults the value of the console environment variable, bus_probe_algorithm, to "new." This is the recommended setting for OpenVMS Version V6.2 and subsequent releases.

The "new" setting can result in a difference between the console controller letter assignments and the operating system controller letter assignment, especially if any PCI/PCI bridges are present in the system. The console environment variable, bus_probe_algorithm, can be reset to "old" to restore the existing controller letter assignment.

After you change this environment variable, you must power-cycle, or reset the system to ensure that the environment variable has taken effect. It is not possible to use the "new" setting for prior versions of the operating system.

1.5.1.2 Graphics Support

V6.2

The following graphics support enhancements are included in this release:

- This release supports the QVision PCI card (1280/P) for AlphaServer 1000, 2000, and 2100 series systems.
- This release supports the AlphaServer 1000 with embedded Cirrus system graphics.

  This support consists of a Cirrus CL-GD5422/24 VGA controller integrated on the system motherboard with supported resolutions of 800x600x256 and 640x480x256. (Refer to Section 1.5.2.5 for restrictions on Cirrus system graphics.)

1.5.2 Problems and Restrictions

This section describes hardware problems and restrictions pertaining to OpenVMS Alpha systems.

1.5.2.1 Adapter Support

V6.2

If your AlphaStation 200 or 400 series computer is configured with multiple ZLX–E* graphics adapters (a multihead system), you must turn the computer off and then on (power-cycle) before rebooting.

1.5.2.2 Bells on AlphaStation 400 Systems

V6.2

DECwindows now supports the bell function on AlphaStation 400 systems. There is no hardware support for adjusting the volume. However, you can use the Bell Volume slider in the Keyboard Options dialog box under the Session Manager Options menu to adjust the frequency of the tone. You can use the Keyclick Volume slider in the same dialog box to adjust the duration.

1.5.2.3 DECterm Startup

V6.2

DECterm windows might not start up properly when invoked using the Automatic Startup option in the Session Manager.

To correct this problem, set DECterm to run as a detached process by default. Follow these steps:

1. Click on the Applications menu in the Session Manager.
2. Click on the horizontal ellipsis (...) next to the DECterm option.
3. Click on the Detached option in the Run Application section of the DECterm Qualifiers dialog box.
4. Click on Save to make this change the default.

Note: This problem is fixed in DECwindows Motif for OpenVMS Version 1.2–3. However, if you have saved a private profile file (DECW$USER_DEFAULTS:VUE$PROFILE.VUE$DAT) from an earlier version, you may still have to apply the correction described in this note.

1.5.2.4 Ethernet Wire Settings

V6.2

Note the following:

- Because the EWA0_MODE parameter AUTOSENSING does not currently work on AlphaStation 200 and 400 series systems, you must use one of the following console commands to manually set your system to either AUI (thickwire or ThinWire) or TWISTED (twisted pair or 10Base-T):

  ```
  >>> SET EWA0_MODE AUI
  >>> SET EWA0_MODE TWISTED
  ```

- If you use the SET EWA0_MODE AUI setting, be sure the specified mode matches the setting (thickwire or ThinWire) for the jumper on the PCI Ethernet card.

1.5.2.5 Graphics Support

V6.2

Note the following restrictions on graphics support:

- ZLX–E* graphics are supported on AlphaStation 200 and 400 series systems only.
- No ISA graphics adapters are supported.
- On all AlphaStation series computers with graphics, be sure the console is set correctly by entering the following command at the console prompt:

  ```
  >>> SET CONSOLE GRAPHICS
  ```

- Note these restrictions on the AlphaServer 1000 with embedded Cirrus system graphics:
  - The Cirrus CL-GD5422/24 VGA controller provides no accelerated drawing.

- The AlphaServer 1000 motherboard jumper that enables the Cirrus chip must be installed and all EISA bus graphics adapters must be removed prior to using this option.
- Multi-head graphics support is not provided if the Cirrus embedded graphics device option is used.
- The embedded Cirrus graphics support is intended for limited use and may not be suitable for general use as a Motif display. DECterm text scrolling may be slow and the limited display size may preclude the use of certain applications.
- Because the Cirrus embedded graphics controller on AlphaServer 1000 supports a resolution (800x600) that is different from the 1024x768 resolution supported by DECwindows applications, some application screen displays may extend beyond the image area of the screen. The workaround for this problem is to add scroll bars to those applications by doing the following:

  1. While in a DECwindows session, add the following resource definition to the file [SYSMGR]DECW$XDEFAULTS.DAT:

     ```
     *DXmfitToScreenPolicy: AS_NEEDED
     ```

  2. After editing (or creating) the resource file, end the DECwindows session.

  When you log back in to the system, the DECwindows applications that have long screen displays will have scroll bars.
- If the display performance is not adequate for your needs, you can use optional EISA or PCI bus VGA graphics controllers instead. However, if you install optional graphics, you must disable the embedded Cirrus graphics by removing the jumper from the motherboard.

1.5.2.6 HALT/CONTINUE Restriction

V6.2 On AlphaStation 400 series systems, use Ctrl/P to perform HALT/CONTINUE operations instead of pressing the Reset button. Pressing the Reset button resets the mouse and keyboard.

1.5.2.7 KFESA Port Reinitializing Message

V6.2 On AlphaStation 1000 series systems, if a KFESA DSSI disk controller is installed, you may see several instances of the following message while booting:

```
%PAA0, Port is Reinitializing
```

You can ignore this message. If you do not want this message to display, enter the console INIT command before entering the boot command.

1.5.2.8 RX23 Disks Supported Instead of RX26 Disks

V6.2 AlphaStation 200 and 400 and AlphaServer 1000 series systems support RX23 disks but not RX26 disks. However, the driver for the diskette device, DVDRIVER, currently identifies all diskette drives as RX26. As a result, the following command produces a display that identifies the diskette drive as an RX26:

```
$ SHOW DEVICE/FULL DV
```

Note that the RX23 is capable of supporting double density (DD) diskettes and high density (HD) diskettes. It does not support extended density (ED) diskettes.

1.5.2.9 SCSI Controller Restriction

V6.2

The Adaptec 1740/1742 SCSI controller (PB2HA–SA) is not supported on AlphaServer 2100 systems having more than 1 gigabyte (GB) of memory. If the controller is connected to such a system, the following message appears on the operator's console:

```
%PKJDRVR-E- The direct DMA window does not map all of memory. Port is going OFF LINE.
```

1.5.2.10 Window Corruption

V6.2

On AlphaStation 200 and 400 series systems, some windows will become corrupted (will not repaint correctly) after an operator message is displayed. This occurs when either of the following conditions exists:

- A window has backing store enabled.
- A window is occluded by a window with SaveUnders enabled. (Menus with DECwindows Motif for OpenVMS Version 1.2 and later enable SaveUnders.)

There are two workarounds for this problem:

- Shrink the corrupted window to an icon and then restore it.
- Refresh the screen by entering the following command:

```
$ MC DECW$UTILS:XREFRESH
```

The DECwindows Motif 1.2-3 kit provides a Console Window application to prevent this problem when using DECwindows. Refer to the *DECwindows Motif Version 1.2–3 for OpenVMS Release Notes* and Section 3.11.2.1 for more information about the Console Window application. ♦

1.6 Ethernet and FDDI Controllers

This section contains release notes that pertain to Ethernet and FDDI controllers.

1.6.1 Problems and Restrictions

This section describes problems and restrictions that pertain to certain Ethernet and FDDI controllers.

1.6.1.1 Multiple Controllers Configured But Not All Attached to a Wire (Alpha Only)

V6.2

Alpha

If you have multiple Ethernet and FDDI controllers configured on your system, and not all of them are attached to a wire, you might experience problems. Problems can range from not being able to access all the services available on the network, if you have four or more controllers configured, to a system crash.

To avoid these problems, specify only the controllers that are attached to the wire. Digital recommends that you do this by first editing your SYS$STARTUP:ESS$LAST_STARTUP.DAT data file to specify only the controllers that are attached and then restarting your system.

With certain controller configurations, if you do not specify only the controllers that are attached, your system might crash when you issue the following command sequence:

```
$ MC ESS$LASTCP
LASTCP> STOP
```

An example of how to edit the SYS$STARTUP:ESS$LAST_STARTUP.DAT file follows. The unedited file is shown first, followed by an edited file.

```
!++
! This file will be used to set the appropriate LASTCP qualifiers. The following
! LASTCP qualifiers: ALL_CONTROLLERS, CHECKSUM, TRANSMIT_QUOTA, or SLOW_MODE
! can be set by using the following statement format:
! LASTCP qualifier = 1 to enable   e.g. SLOW_MODE = 1 enables  SLOW_MODE
! LASTCP qualifier = 0 to disable  e.g. SLOW_MODE = 0 disables SLOW_MODE
! The remaining LASTCP qualifiers will require the appropriate value settings.
! DEVICE          = (list-of-devices)
! TIMEOUT    = n    minimum interval in seconds
! CIRCUIT_MAXIMUM = n    maximum number of nodes
! GROUP           = n    Group number
! NODE_NAME       = name   Node name
! CONTROLLERS     = ([{controller letter,}...]) Controller list
! TRANSMIT_QUOTA  = n    Number of transmit buffers
!--
ALL_CONTROLLERS = ON
```

The edited SYS$STARTUP:ESS$LAST_STARTUP.DAT file follows. This example assumes you have ESA, ETA, EXA, EZA controllers configured on your system and that only the ESA controller is attached to the Ethernet wire.

```
!++
! This file will be used to set the appropriate LASTCP qualifiers. The following
! LASTCP qualifiers: ALL_CONTROLLERS, CHECKSUM, TRANSMIT_QUOTA, or SLOW_MODE
! can be set by using the following statement format:
! LASTCP qualifier = 1 to enable   e.g. SLOW_MODE = 1 enables  SLOW_MODE
! LASTCP qualifier = 0 to disable  e.g. SLOW_MODE = 0 disables SLOW_MODE
! The remaining LASTCP qualifiers will require the appropriate value settings.
! DEVICE          = (list-of-devices)
! TIMEOUT    = n    minimum interval in seconds
! CIRCUIT_MAXIMUM = n    maximum number of nodes
! GROUP           = n    Group number
! NODE_NAME       = name   Node name
! CONTROLLERS     = ([{controller letter,}...]) Controller list
! TRANSMIT_QUOTA  = n    Number of transmit buffers
!--
ALL_CONTROLLERS = OFF
DEVICE = (ESA)
```

Note

The default ESS$LAST_STARTUP.DAT file is stored in SYS$COMMON:[SYS$STARTUP]. You might want to put the edited file in SYS$SPECIFIC:[SYS$STARTUP]. Otherwise, other system roots might be affected. ♦

1.7 RF73 and Other RF*nn* DSSI Disk Devices

Notes in this section pertain to the RF31T, RF31T+, RF35, RF35+, RF73, and RF74 DSSI disk devices.

1.7.1 Problems and Restrictions

This section describes a problem found in certain RF31T, RF31T+, RF35, RF35+, RF73, and RF74 DSSI disk devices.

1.7.1.1 RF73 and Other RF*nn* DSSI Disk Devices and Controller Memory Errors

A problem exists with the microcode for earlier versions of RF31T, RF31T+, RF35, RF35+, RF73, and RF74 DSSI disk devices that can cause data loss. The problem can occur when reading data from one of these devices when the device has had a Controller Memory error (also known as an error detection and correction (EDC) error). The error could have been induced by a virtual circuit closure or faulty hardware.

Digital advises customers with any of these devices to check their microcode revision levels. If the microcode revision levels are lower than the numbers shown in Table 1–1, Digital recommends that you update the microcode. The microcode for all models, except the RF31T, RF31T+, and RF35+, is provided on the OpenVMS Version 6.2 binary distribution CD–ROM.

The RF_VERS utility, a utility program that displays the microcode revision level of the DSSI disk devices, is also provided on the CD–ROM. Instructions for using the utility program and for updating the microcode are provided in this section.

Note

If you have an RF31T, RF31T+, or RF35+ disk drive with a version of microcode that is not error-free (see Table 1–1), and if you have a support contract, contact your Digital support representative. Otherwise, contact your Digital account representative or your authorized reseller.

The earliest supportable revision levels of the DSSI disk microcode are shown in Table 1–1.

Table 1–1 Error-Free Microcode Revision Levels

Device Type	Minimum Level with Error-Free Microcode
RF31T	T387E
RF31T+	T387E
RF35	T392D
RF35+	T392D
RF36	V427P
RF73	T392D
RF74	V427P

To display the microcode version level of your DSSI disk devices, perform the following steps:

1. Log in to the SYSTEM account or another account that has the CMKRNL, DIAGNOSE, and SYSPRV privileges.
2. Issue the following commands:

```
$ SET PROCESS /PRIVILEGE=(DIAGNOSE,CMKRNL,SYSPRV)
$ SHOW DEVICE FYA0:
```

VAX

On VAX systems, if the SHOW DEVICE command produces an error, issue the following commands:

```
$ RUN SYS$SYSTEM:SYSGEN
        SYSGEN>  CONN FYA0/NOADAP
        SYSGEN>  ^Z ♦
```

Alpha

On Alpha systems, if the SHOW DEVICE command produces an error, issue the following commands:

```
$ RUN SYS$SYSTEM:SYSMAN
        SYSMAN> IO CONNECT FYA0: /NOADAP
        SYSGEN>  ^Z ♦
```

3. On VAX and Alpha systems, issue the following command:

```
$ RUN SYS$ETC:RF_VERS.EXE
```

The following is an example of the display produced by the RF_VERS utility:

```
Program Name:   RF_VERS
Revision Level: V1.2s

NOTICE: This program does not currently support the RF72 or any
        HSDxx controllers. See next version for support.

DSSI disks currently on this system as seen by RF_VERS

Device            Node        Status            Hardware    Firmware
Name              Name                          Type        Version

_$22$DIA7:        R4JL2I      mounted           RF73        T387A
_$22$DIA6:        R4I0BG      mounted           RF73        T387A
_$22$DIA8:        R4XLWE      mounted           RF73        T387A
_$22$DIA2:        R4FCZK      mounted           RF73        T387A
_$22$DIA3:        R4CKCG      mounted           RF73        T387A
_$22$DIA4:        R4ZKUE      mounted           RF73        T387A
_$22$DIA9:        R4GYYI      mounted           RF73        T387A
_$22$DIA1:        R4XRYI      mounted           RF73        T387A
```

To update the microcode in your device, use the appropriate command for your device and platform from Table 1–2.

Caution

Back up the disk before updating the microcode.

Table 1–2 Commands for Updating Microcode in Certain DSSI Disk Devices

Device Type	Platform	Command
RF35	Alpha	$RUN SYS$ETC:RF35_T392F_DEC_ALPHA.EXE
RF35	VAX	$RUN SYS$ETC:RF35_T392F_DEC.EXE
RF36	Alpha	$RUN SYS$ETC:RF36_V427P_DEC_ALPHA.EXE
RF36	VAX	$RUN SYS$ETC:RF36_V427P_DEC.EXE
RF73	Alpha	$RUN SYS$ETC:RF73_T392F_DEC_ALPHA.EXE
RF73	VAX	$RUN SYS$ETC:RF73_T392F_DEC.EXE
RF74	Alpha	$RUN SYS$ETC:RF74_V427P_DEC_ALPHA.EXE
RF74	VAX	$RUN SYS$ETC:RF74_V427P_DEC.EXE

Caution

Do not delete SCSI_INFO.EXE, RF_VERS.EXE, or any of the files listed in Table 1–2. If these files are deleted, VMSKITBLD.COM (on VAX) will not be able to find them. Similarly, on Alpha systems, the PRODUCT INSTALL commands in AXPVMS$PCSI_INSTALL and AXPVMS$PCSI_INSTALL_MIN will fail.

1.8 Hardware Information: VAX 4000 Series Computers (VAX Only)

VAX

This section contains notes pertaining to VAX 4000 Series Computers.

1.8.1 Changes and Enhancements

V6.1

The following notes supplement the information contained in *OpenVMS Upgrade /Installation: VAX 4000 Series, MicroVAX, VAXstation, and VAXserver 32/33/34 /35/36/38/3900 Series*:

- For certain VAX 4000 series systems with embedded DSSI buses, you can specify a DSSI bus number from 0 to 3 (rather than only 0 or 1).
- After you start the DUP Driver utility, you can change the DSSI node name by entering the following command at the PARAMS> prompt. For example:

```
PARAMS> SET NODE <BARNEY> ♦
```

2
General User Release Notes

This chapter provides information for all users of the OpenVMS operating system. It includes information about commonly used commands and utilities.

For information about new features included in this version of the software, refer to *OpenVMS Version 6.2 New Features Manual*.

2.1 Batch and Print Queues

This section contains release notes of general interest to most users of batch and print queue operations. Notes for this facility are also included in Chapter 3 and Chapter 4.

2.1.1 Corrections

This section describes corrections to batch and print queue software.

2.1.1.1 File Search Qualifiers

V6.2

In previous releases, a problem existed when you entered a BATCH or PRINT command with one of the following search qualifiers:

/BY_OWNER
/CONFIRM
/EXCLUDE
/SINCE
/BEFORE

If no files met the selection criteria, an empty file list was sent to the Job Controller and the following error message was issued:

```
%SUBMIT-F-CREJOB, error creating job
-JBC-F-MISREQPAR, missing required parameter
```

With this release of OpenVMS, the Job Controller is not called if there is an empty file list (that is, if no files meet the selection criteria) and the following error message is issued:

```
%SUBMIT-F-CREJOB, error creating job
-LIB-W-FILFAIMAT, file failed to match selection criteria
```

2.1.1.2 Print Job Alignment

V6.2

In prior releases, there was a problem aligning print jobs. If the end of a print job was reached during alignment, the job would complete normally; however, if the queue characteristics or the PRINT command did not have the /RETAIN qualifier, the job would disappear.

This problem has been fixed.

2.2 DCL Commands

This section contains release notes related to the DIGITAL Command Language (DCL) for this release of the OpenVMS operating system.

2.2.1 Problems and Restrictions

The notes in this section describe problems and restrictions pertaining to DCL commands.

2.2.1.1 SET PROCESS/NOAUTO_UNSHELVE Command in Cluster Environment

V6.1

The command SET PROCESS/NOAUTO_UNSHELVE as implemented in Version 6.1 does not support operations across the cluster. It can be issued only for a process on the same node, including as the default case, the process from which the command is issued.

The /IDENTIFICATION=pid qualifier is supported, but only when the target process is on the same node as the process where the command is issued.

2.2.1.2 SET PROCESS/SUSPEND=KERNEL/ID= Command in Cluster Environment

VAX V6.0
Alpha V1.5

When you issue the SET PROCESS/SUSPEND=KERNEL/ID= command in a cluster environment, the KERNEL keyword is ignored if the target process and the current process reside on different cluster nodes. As a result, process suspension is handled as if you had specified the SUPERVISOR keyword (the default).

This is caused by a problem with the $SUSPND system service, as discussed in Section 4.26.2.2. Digital expects to fix this problem in a future version of the OpenVMS operating system.

2.2.2 Corrections

The notes in this section describe software corrections for problems associated with DCL commands.

2.2.2.1 DELETE/QUEUE/MANAGER Command

V6.2

In previous versions, the following error message (which was generated when the required /NAME_OF_MANAGER=name qualifier was omitted from the DELETE /QUEUE/MANAGER command) displayed an incomplete qualifier name:

```
%DCL-W-CONFLICT, illegal combination of command elements - check documentation
\NONAME\
```

This problem has been corrected. The error message now reads:

```
%DCL-W-CONFLICT, illegal combination of command elements - check documentation
\NONAME_OF_MANAGER\
```

2.2.2.2 INITIALIZE/QUEUE Command

V6.2

Previously, you could specify negative values for the working set qualifiers /WSDEFAULT, /WSEXTENT, and /WSQUOTA even though such values were meaningless.

This problem has been corrected. Negative values are no longer accepted. If you specify a negative value, you get an error message.

2.2.2.3 PRINT/DELETE Command

V6.2 Previously, if you submitted a print job with the PRINT/DELETE command, the file associated with a print job was deleted even if the print operation was unsuccessful.

This problem has been corrected. Files marked for deletion with the /DELETE qualifier are now deleted only if the print job completes successfully.

2.2.2.4 PRINT/FORM Command on LAT Printers

V6.2 A problem existed that caused the LATSYM to crash when the /FORM qualifier specifying a WIDTH of 255 was used to print a file containing an OSC escape sequence on a LAT printer.

This problem has been corrected.

2.2.2.5 SET QUEUE/DESCRIPTION Command

V6.2 In previous versions, entering the SET QUEUE/DESCRIPTION command with a description string above a certain limit would cause the SHOW QUEUE command to display some extraneous characters.

This problem has been corrected.

2.2.2.6 SET QUEUE/NODESCRIPTION Command

V6.2 In previous versions of the operating system, the queue description was not deleted when the DCL command SET QUEUE/NODESCRIPTION was executed. This problem has been corrected, and the description is now deleted when the SET QUEUE/NODESCRIPTION command is executed.

2.2.2.7 SHOW SYSTEM Command

V6.2 Previously, if you entered the SHOW SYSTEM command on a VAX node, the operating system was displayed as `VAX/VMS`. In a mixed-architecture VMScluster, this difference could cause problems for applications that parse the SHOW SYSTEM display.

This problem has been corrected. The operating system for both OpenVMS Alpha and OpenVMS VAX is identified in the display as `OpenVMS`.

2.2.2.8 START/QUEUE Command

V6.2 In previous versions, there were problems associated with the START/QUEUE command.

When you used the START/QUEUE/SEARCH="text string" command to restart a paused print job, the print job did not start on the correct page (that is, the page containing the search text). Instead, the print job started a few lines before the actual page containing the text. This problem has been corrected.

When you used the START/QUEUE/BACKWARD=*n* command to restart a queue after a paper jam and the paper was at top-of-form, the print job would start with the last line of the page before page *n*. This misalignment would continue for the rest of the print job. This problem has been corrected.

2.2.2.9 SUBMIT Command

V6.2

If a command procedure was submitted under the following conditions, a problem caused a batch process to overwrite with a log file:

- A logical name for the command procedure was used in the SUBMIT command.
- The /NAME qualifier was omitted.
- The /LOG qualifier was defaulted or omitted.

This problem has been corrected.

2.2.2.10 SUBMIT/NAME and SET ENTRY/NAME Commands

V6.2

In previous versions, when you used a SUBMIT/NAME command or a SET ENTRY/NAME command on a print queue, the following error message was returned:

```
%JBC-I-ITMREMOVED, meaningless items were removed from request
```

This message was returned even though there were no meaningless items in the command.

This problem has been fixed.

2.2.2.11 SUBMIT/NAME/USER Command

V6.2

In previous versions, the combined use of the SUBMIT command qualifiers /NAME and /USER caused the log file to be created in the wrong directory. For example:

- If the following command was used to submit a command file for another user, the log file would be created in the account specified by the /USER qualifier:

  ```
  $ SUBMIT SOME_WORK.COM /KEEP /NOPRINT /USER=DIMPLE
  ```

  In this example, the log file SOME_WORK.LOG would be created in the directory [DIMPLE]. This is correct.

- However, if the following command—containing both the /USER and /NAME qualifiers—was used to submit that same file, the log file would be created in the default login directory of the user submitting the file:

  ```
  $ SUBMIT SOME_WORK.COM /KEEP /NOPRINT /USER=DIMPLE /NAME=BOUNCER
  ```

  If the file was submitted from the account VOODOO, then the log file BOUNCER.LOG would be created in the wrong directory [VOODOO].

This problem has been corrected. The log file is now created in the right directory.

2.3 DECnet Layered Product

This section contains release notes pertaining to the DECnet layered product.

2.3.1 Changes and Enhancements

The release notes in this section describe changes and enhancements to the DECnet layered product on OpenVMS Version 6.2.

2.3.1.1 DECnet Phase IV Superseded by DECnet/OSI

V6.2

The DECnet/OSI layered product supersedes the DECnet Phase IV product distributed with the OpenVMS operating system. DECnet Phase IV still ships with the OpenVMS operating system, but it will be retired in a future release.

DECnet/OSI (DECnet Phase V) was first released on OpenVMS in 1991. As with all previous releases of DECnet products, there is full backwards compatibility with the previous version. As such, customers can continue to use Phase IV network protocols and applications with DECnet/OSI.

The current release of DECnet/OSI provides a suite of tools for ease of upgrade and ease of use as customers make the transition to the new environment. Future releases will incorporate additional enhancements and improvements based on customer feedback and needs. Some of these improvements include making NCL network management easier to use and increasing the size of the local name space to accommodate larger networks with local name services.

DECnet/OSI for OpenVMS on VAX and Alpha platforms provides the functionality of DECnet plus support of the international OSI protocols. For more information about upgrading to DECnet/OSI, see Section 1.1.2.1.

2.4 DECserver

This section contains notes pertaining to DECservers.

2.4.1 Problems and Restrictions

This section describes a restriction with the baud rate when using LAT.

2.4.1.1 Baud Rate Using LAT

V6.1

The DECserver 90TL and the DECserver 90M support port speeds of 57600. The DECserver 700 terminal controller supports 57600, 76800, and 115200 bauds. However, when the DECserver communicates with OpenVMS using LAT, a restriction allows LAT to report speeds only up to 57600. This restriction will be addressed in a future release.

2.5 DECTPU

This section contains notes pertaining to the DEC Text Processing Utility (DECTPU).

2.5.1 Problems and Restrictions

The notes in this section describe DECTPU problems and restrictions.

2.5.1.1 Motif Widget Context Help Built-In

V1.0

The following built-in, which should enter Motif context-sensitive help mode, is disabled because of a problem in the Motif toolkit:

```
SET (WIDGET_CONTEXT_HELP, widget_variable, {on|1|off|0})
```

The mouse pointer changes to a question mark, and DECTPU waits for you to select a widget by clicking on MB1. DECTPU then executes the help callback of the selected widget (or of its parent if the selected widget has no help callback). The widget_variable is the widget within which the modal help interaction will occur, usually the top-level widget returned from the GET_INFO (SCREEN, "widget") built-in. The last parameter confines the question mark pointer to the specified widget if ON or 1, and does not confine the pointer if OFF or 0.

2.6 DECwindows Motif for OpenVMS

This section contains two release notes of general interest pertaining to the DECwindows Motif for OpenVMS layered product. All other notes for this product are included in Section 3.11 and related sections.

2.6.1 Changes and Enhancements

This section includes notes about support for the the DECwindows Motif for OpenVMS layered product.

2.6.1.1 Support for DECwindows Motif Version 1.1 for OpenVMS

V6.2

OpenVMS Version 6.2 is the final version of the OpenVMS operating system to support DECwindows Motif Version 1.1 for OpenVMS. Future versions of OpenVMS will require that DECwindows Motif Version 1.2 or later be installed.

2.6.1.2 NCSA Mosaic Browser: TCP/IP Support Options

V6.2

DECwindows Motif Version 1.2–3 includes the NCSA Mosaic web browser. The *DECwindows Motif Version 1.2–3 for OpenVMS Release Notes* state that the NCSA Mosaic browser requires that the TCP/IP Services for OpenVMS product (UCX) be installed on your system. Note that TCP/IP support can alternatively be provided by PathWay for OpenVMS (Wollongong) or TCPware (Process Software).

2.7 Image Activator

This section contains a release note pertaining to the Image Activator.

2.7.1 Changes and Enhancements

This section describes an enhancement to the Image Activator.

2.7.1.1 Message Returned on Attempt to Run an Alpha Image on a VAX System (VAX Only)

V6.2

VAX

The Image Activator has been enhanced to return this error message if you attempt to run an OpenVMS Alpha image on an OpenVMS VAX system:

```
$ RUN alpha_image.exe
%DCL-W-ACTIMAGE, error activating image ALPHA_IMAGE
-CLI-E-IMGNAME, image file _$1$DKA0:[MY_DIR]ALPHA_IMAGE;1
-IMGACT-F-NOTVAXIMG, image is not an OpenVMS VAX image ♦
```

2.8 Sort/Merge Utility

This section contains release notes pertaining to the OpenVMS Sort/Merge utility (SORT/MERGE).

2.8.1 Changes and Enhancements

This section describes changes and enhancements to SORT/MERGE.

2.8.1.1 /FIELD Qualifier

V6.2

The VALUE:n parameter has been added to the specification file /FIELD qualifier:

/FIELD

The VALUE:n parameter of the /FIELD qualifier allows you to define a constant field and assign it a value of any valid Sort/Merge data type. You can use this constant field in specification file /KEY, /DATA, and /CONDITION statements.

Table 2–1 describes the /FIELD formats.

Table 2–1 Formats of the /FIELD Qualifier for SORT/MERGE Specification Files

Qualifier	Default
/FIELD=(NAME=field-name,POSITION:n, SIZE:n[,DIGITS:n][,data-type])	None
/FIELD=(NAME=field-name,VALUE:n, SIZE:n[,DIGITS:n][,data-type])	None

If you specify VALUE:n, do not specify POSITION:n. The VALUE parameter is a constant and is not part of an input record.

The VALUE:n parameter was previously undocumented, but it is available in OpenVMS VAX releases beginning with Version 5.1 and in OpenVMS Alpha releases beginning with Version 1.0.

3
System Management Release Notes

This chapter contains information that applies to system maintenance and management, performance management, and networking.

For information about new features included in this version of the software, refer to *OpenVMS Version 6.2 New Features Manual*.

3.1 Authorize Utility (AUTHORIZE)

This section contains release notes pertaining to the Authorize utility.

3.1.1 Changes and Enhancements

This section notes a change to the working set quota (WSQUOTA).

3.1.1.1 WSQUOTA Maximum Value Is 64K Byte Pages

V6.2

The working set quota (WSQUOTA) maximum value is 65536 (64K byte) pages.

You may be able to set a higher value, but the maximum will be enforced at the time the process is created.

3.2 AUTOGEN

This section contains release notes pertaining to AUTOGEN.

3.2.1 Changes and Enhancements

This section describes changes and enhancements to AUTOGEN on OpenVMS VAX systems.

3.2.1.1 AGEN$PARAMS.REPORT: Turning Off Logging (Alpha Only)

V6.2

Alpha

The contents of MODPARAMS.DAT are evaluated as DCL statements, and you can make assignments to symbols with names that are not SYSGEN parameters (for example, "scratch" variables or conditional assignments based on other values). Traditionally, every such assignment is logged in AGEN$PARAMS.REPORT, sometimes creating a large file with many logging statements that do not interest users.

Starting with OpenVMS Alpha Version 6.2, you can designate any assignments that you prefer not to log in AGEN$PARAMS.REPORT by prefixing every such assignment with a dollar sign ($). When AUTOGEN encounters a MODPARAMS.DAT record beginning with a $, it does not check the list of known SYSGEN parameters and does not log this record to AGEN$PARAMS.REPORT. ♦

3.2.1.2 CTLPAGES Parameter Value (VAX Only)

V6.2

VAX

The value of the parameter CTLPAGES is now computed, based on the default value and on any site-specific changes to LNMPHASHTBL (as well as direct modifications of the value of CTLPAGES). ♦

3.2.1.3 Dump File Size (VAX Only)

V6.2

VAX

As in previous releases, the software dumps the system to SYSDUMP.DMP if the file is present — even if the SYSGEN parameter SAVEDUMP is set to 1 to indicate that system dumps be dumped to PAGEFILE.SYS instead.

Previously, AUTOGEN would not compute and adjust the size of the dump file if one was present. Starting with Version 6.2, AUTOGEN calculates the proper size for the dump file if a dump file exists. If execution proceeds through the GENFILES phase, AUTOGEN might even enlarge the dump file. ♦

3.2.1.4 Retired Parameters (VAX Only)

V6.2

VAX

AUTOGEN has a list of parameters that have been retired in this version as well as earlier versions of OpenVMS. When one of them is encountered in MODPARAMS.DAT, AUTOGEN will make the symbol assignment and enter an informational message in the report file, just as it does for "scratch" variables.

The current list of retired parameters is as follows:

CRDENABLE
IRPCOUNT, IRPCOUNTV
LRPCOUNT, LRPCOUNTV, LRPMIN, LRPSIZE
SRPCOUNT, SRPCOUNTV, SRPMIN, SRPSIZE
TTY_OWNER, TTY_PROT
AWSMIN_SYS, AWSTIME_SYS
PFRATL_SYS, PFRATH_SYS
WSINC_SYS, WSDEC_SYS
QUORUM ♦

3.2.1.5 SYSMWCNT Computations (VAX Only)

V6.2

VAX

The computation of SYSMWCNT (working set for the operating system, in pages) now uses the first 8192 pages of SPTREQ (additional system page table entries), instead of a much smaller subset.

SYSMWCNT for MicroVAX systems is still capped at 10% of physical memory, but the computation no longer includes the restriction of reducing by 75% the amount that exceeded 1000 pages.

These two changes can be expected to increase SYSMWCNT by 200 to 500 pages on MicroVAX systems, and by 100 to 200 pages on other VAX systems. ♦

3.2.1.6 WSMAX Computations (VAX Only)

V6.2

VAX

The computation of WSMAX is no longer linear but resembles a logarithmic curve. Instead of simply computing WSMAX as a quarter of physical memory, it is now computed as a quarter of the first 32 MB, plus a sixteenth of the memory from 32 to 256 MB, plus a sixty-fourth of the memory (if any) above 256 MB.

This is intended to assist managers of systems that host large numbers of users whose working sets are not large. Systems whose user bases consist of a small number of users (or processes) that require large amounts of physical memory (for example, simulations) might need to set MIN_WSMAX to a value that satisfies the requirements of those processes. ♦

3.2.2 Problems and Restrictions

This section describes known AUTOGEN problems and restrictions.

3.2.2.1 Full Memory Dumps (Alpha Only)

V6.2

Alpha

OpenVMS Alpha Version 6.2 does not perform full memory dumps (the system parameter DUMPSTYLE set to 0 or 2) on systems with over 4 gigabytes of memory. Selective dumps (the system parameter DUMPSTYLE set to 1 or 3) are supported for all memory sizes.

The default action by AUTOGEN is to set DUMPSTYLE to 1. This is the recommended setting for all systems. ♦

3.3 Backup Utility

Notes in this section pertain to the Backup utility.

3.3.1 Problems and Restrictions

This section describes known problems and restrictions for the Backup utility.

3.3.1.1 Image and Incremental Backups

V6.2

The first time you back up a disk, you must perform an image backup using the BACKUP/IMAGE/RECORD command before you perform regular incremental backups. The image backup saves a copy of the entire disk and marks each file as being saved. Subsequent incremental backups assume an image backup has been performed and, therefore, save only new or modified files.

If an image backup was not performed first, the incremental backups save more files than might be necessary to ensure that an incremental restore will be successful.

3.3.1.2 Incremental Backups Using PATHWORKS for OpenVMS Servers

V6.2

An incompatibility between the operating procedures of the PATHWORKS for OpenVMS Macintosh server and OpenVMS incremental backup operations can cause BACKUP to save entire disks or directory structures, including subdirectories and files.

A recent change to fix other problems now causes Backup to detect whether a directory file has been modified since the date indicated by the Backup Date field in the file header. If a directory file has been modified, all subdirectories and files of that directory are saved for possible later restore operations. Updating the modification date of directory files is unusual for OpenVMS systems, but it can happen, for example, if you rename a directory file from one location to another.

By contrast, the PATHWORKS Macintosh server maintains the modification date of directory files for Macintosh users; that is, it updates the modification date for each directory change, file creation, and file deletion.

Thus, an incremental backup of a disk where PATHWORKS is used to serve files to Macintosh users may result in saving the entire disk or entire directories (including their subdirectories and files) instead of just the user files that were created or modified since the last incremental backup operation.

This incompatibility will be addressed in a future version of OpenVMS.

For now, you can avoid needless saving of files by performing a dummy BACKUP /RECORD operation on all directory files *immediately* before performing the incremental backup. The following example illustrates this workaround:

```
$ BACKUP/RECORD/IGNORE=(INTERLOCK) -
_$ disk:[000000...]*.DIR;* -
_$ NLA0:DUMMY.BCK/SAVE/NOCRC/GROUP_SIZE=0
$
$ BACKUP/VERIFY/FAST/RECORD/IGNORE=(INTERLOCK) -
_$ /NOASSIST/COMMENT="Incremental backup of DISK:" -
_$ disk:[000000...]*.*;*/SINCE=BACKUP -
_$ tape:incr.bck/LABEL=incr/SAVE
```

In this example, the first BACKUP command performs the dummy backup operation and the second performs the actual incremental backup. The first command updates the Backup Date field for all the directory files. Specifying the null output device [000000...] causes no saveset file to actually be written. Since no file information need be retained from this operation, the /NOCRC and /GROUP_SIZE=0 qualifiers are specified to avoid CRC and XOR block calculation.

3.3.1.3 Warning Issued on ANALYZE/DISK Operation

V6.2

An ANALYZE/DISK operation performed immediately after a BACKUP/IMAGE restore of a disk might result in a warning message similar to the following:

```
%ANALDISK-W-ALLOCCLR, blocks incorrectly marked allocated
       LBN 97 to 105, RVN 1
```

This can be caused by attempting to perform a BACKUP/IMAGE restore operation where alias file entries are restored as separate (primary) file entries. (The primary file, which uses the same file header but allocates different data storage blocks, is also restored.)

Note that there is no BACKUP error or loss of data.

This problem will be addressed in a future version of OpenVMS.

3.3.1.4 CD–ROM Menu System (VAX Only)

V6.1

VAX

Do not use the CD–ROM menu system, which displays when you boot the OpenVMS VAX operating system CD–ROM, to back up user disks. Use it to back up *system* disks only.

When you boot from the SYS1 directory on the distribution CD–ROM, you are booting a writelocked system disk that does not allow paging. Because of this, the system displays error messages similar to the following:

```
%SYSINIT-E, error opening page file, status = 0000025C
%SYSINIT-E, error opening swap file, status = 0000025C
%SYSINIT, primary PAGEFILE.SYS not found; system initialization continuing
%SYSINIT, no dump file - error log buffers not saved
%SYSINIT-E, error mounting system device, status = 00000F64
```

These messages are normal. The lack of page and swap files does not affect most operations.

If you back up large user disks, BACKUP may need to page and the operation could fail. Use online BACKUP to back up user disks.

This can also occur when you use the CD–ROM menu system to back up large system disks on low memory systems (those with less than 32 MB of memory). If this problem occurs, use standalone BACKUP to back up system disks. ♦

3.3.1.5 Image Backups from an RF73 Disk

V6.1

When performing an image backup from an RF73 disk (or a disk with a cluster size of 4 blocks) to an RF74 disk (or a disk with a cluster size of 7 blocks), the Backup utility does not check the file size when it is allocating space for the file being copied. Therefore, if the file has an allocation greater than the value of the CLUSTER_SIZE attribute established during initialization, the Backup utility will allocate one more cluster size number of blocks to the allocation size even though the actual file size is less than the cluster size. For example, during an image backup, a file that uses 6 blocks and is allocated 8 blocks (which displays as 6/8 on the screen if you enter a DIRECTORY/SIZE=ALL command) shows an increase in its allocation size to 14, instead of 7, after it is copied to the target disk.

As a result of this problem, the following files are copied to the image system disk with a blocks used/allocation size of 6/14 blocks:

SYS$COMMON:[SYS$LDR]LIDRIVER.EXE
SYS$COMMON:[SYS$LDR]LPDRIVER.EXE

This incorrect allocation size causes standalone BACKUP to *fail* on the booted image system disk.

To correct this problem, recopy the two previously listed files to the same directory after the image backup, by using the following command (which also specifies the correct allocation size):

```
$ COPY/ALLOCATION=7 SYS$COMMON:[SYS$LDR]LIDRIVER.EXE SYS$COMMON:[SYS$LDR]
$ COPY/ALLOCATION=7 SYS$COMMON:[SYS$LDR]LPDRIVER.EXE SYS$COMMON:[SYS$LDR]
```

3.3.1.6 Relative File Version -0 Processed Like 0

V6.1

The Backup utility processes relative version -0 as if it were 0, saving the most recent version instead of the earliest version of the file for processing.

Digital expects to correct this problem in a future release.

3.3.1.7 Standalone BACKUP Version 5.3 on Console Media (VAX Only)

V6.1

VAX

OpenVMS VAX Version 6.1 and later is available on 9-track magtape in three forms: magnetic tape only or with standalone BACKUP on either RX50 or RL02 console media. The standalone BACKUP on the console media is VMS Version 5.3; the media is so labeled and identifies itself as Version 5.3 on startup. The standalone BACKUP console media have not been updated for OpenVMS VAX Version 6.1 and later.

OpenVMS VAX Version 6.1 and later is not available with standalone BACKUP on RX01 or TU58 console media. Installations using these console media are expected to have standalone BACKUP console media created from previous versions of the operating system. If you need standalone BACKUP on RX01 or TU58 media and are not in a position to create your own kits before installing Version 6.1 or later, please contact your Digital support representative for assistance.

After you have installed OpenVMS VAX Version 6.1 or later, you can build a Version 6.1 or later standalone BACKUP kit on a disk of your choice or on the RL02 or RX50 console media. You cannot build a Version 6.1 or later standalone BACKUP kit on RX01 or TU58 media. If you use either of the latter, you must retain a kit built from a previously installed version of the operating system. ♦

3.3.1.8 Standalone BACKUP Failure on VAX 4000 Model 300 and TF857 (VAX Only)

V6.0

VAX

If a TF857 and the VAX 4000 Model 300 are connected as part of a tri-hosted DSSI configuration, standalone BACKUP fails to operate properly.

For standalone BACKUP to work properly on VAX 4000 Model 300 from a TF857, it is necessary to either have the TF857 connected to no more that one host or to disable the other two hosts by executing the shutdown procedure. ♦

3.3.1.9 VMS$COMMON.DIR File: Restore Problems

VAX V6.0
Alpha V6.1

On an OpenVMS system disk, the file [SYSx]SYSCOMMON.DIR is an alias directory of the file [000000]VMS$COMMON.DIR. This means that both files point to the same file header. Prior to OpenVMS VAX Version 5.5-2 and OpenVMS Alpha Version 1.5, BACKUP did not properly restore the VMS$COMMON.DIR file. Although this does not affect the system disk, it might produce errors with DIGITAL Command Language (DCL) lexical functions.

OpenVMS VAX Version 5.5-2 and OpenVMS Alpha Version 1.5 corrected this problem. However, if you restore image backups created with an old OpenVMS version, the problem can recur.

The symptoms of the problem are different depending on which version of the operating system you are using. If you upgraded from OpenVMS VAX Version 5.5-2 or OpenVMS Alpha Version 1.5 to OpenVMS VAX Version 6.0 or later or OpenVMS Alpha Version 6.1 or later, it is unlikely that your system disk has this problem. However, you should confirm this and correct the problem if necessary.

To restore VMS$COMMON to its proper state, enter the following commands:

```
$ SET DEFAULT DISK:[000000]
$ SET FILE/ENTER=SYSCOMMON.DIR VMS$COMMON.DIR
$ SET FILE/REMOVE VMS$COMMON.DIR;
$ RENAME SYSCOMMON.DIR VMS$COMMON.DIR
```

If you upgraded to OpenVMS Version 6.1 or later from OpenVMS VAX Version 5.5-2 or OpenVMS Alpha Version 1.5 without following this procedure, your system disk could be affected by this problem.

To determine if your system disk has this problem, enter a BACKUP/LIST command to display save set information about the files contained in the VMS$COMMON directory. For example:

```
   .
   .
   .
[000000]VOLSET.SYS;1                                  0  24-SEP-1994 19:31
[]000000.DIR;1                                        1  24-SEP-1994 19:31
[]SYSCOMMON.DIR;1                                     2  24-SEP-1994 19:31
[]SYSLIB.DIR;1                                       18  24-SEP-1994 19:31
[]SYSTEST.DIR;1                                       1  24-SEP-1994 19:31
[]SYSMAINT.DIR;1                                      1  24-SEP-1994 19:31
[]SYSMGR.DIR;1                                        6  24-SEP-1994 19:31
[]SYSHLP.DIR;1                                        6  24-SEP-1994 19:31
[]EXAMPLES.DIR;1                                      1  24-SEP-1994 19:31
[]SYSUPD.DIR;1                                        4  24-SEP-1994 19:31
[]SYSMSG.DIR;1                                        3  24-SEP-1994 19:31
   .
   .
   .
[]SECURITY_AUDIT.AUDIT                                2   3-FEB-1995 15:23
[]SECURITY_AUDIT.AUDIT                               11   3-FEB-1995 15:23
[]BACKUP.EXE;33                                     273   4-FEB-1995 09:37
[]STABACKUP.EXE;9                                   486   4-FEB-1995 09:38
```

If the display lists the files in the VMS$COMMON directory as lost files (files with an empty directory specification as shown in this example), your system disk is affected by this problem. Follow the procedure described above to correct it.

3.3.2 Corrections

This section describes corrections to the OpenVMS Backup utility (BACKUP).

3.3.2.1 BACKUP$_ACCONFLICT Error Message

V6.2

In prior versions of BACKUP, using the /IGNORE=INTERLOCK qualifier would result in BACKUP reporting a BACKUP$_ACCONFLICT message when a file was accessed for WRITE from the local node during the initial opening of the file. If the file was accessed from a remote node or if it was accessed for WRITE from the local node after BACKUP opened the file, a BACKUP$_ACCONFLICT message was not reported, and users would not know that they might have an inaccurate copy of the file.

BACKUP now reports a BACKUP$_ACCONFLICT message if the file is modified while BACKUP is reading the file or if the file is accessed for WRITE on the *local* node when BACKUP is finished reading the file. This correction improves the accuracy of this message.

Note, however, that BACKUP is unable to detect when the file is being accessed for WRITE from a *remote* node after BACKUP is finished reading the file, and no message will be displayed.

3.3.2.2 Boot Block Saved Correctly (VAX Only)

V6.2

VAX

In previous versions of OpenVMS VAX, if an image backup of an Alpha system disk was performed on a VAX system, the boot block was not saved correctly and the user was not notified that there was a problem. If the resulting save set was restored, the Alpha system would not boot.

This problem has been corrected. Now, the boot block is saved correctly and, on a restore operation, the Alpha system reboots. ♦

3.3.2.3 Incremental Restore

V6.2

Previous versions of the OpenVMS Backup utility (BACKUP) would not restore target disk structures as expected during incremental restore operations, resulting in the following problems:

1. Some directories and their contents would be erroneously created or restored (as opposed to deleting or not creating those directories and their contents that were previously deleted or renamed to a new location).
2. In cases where a directory was renamed (moved) to a new location, BACKUP would not be able to detect that the directory was renamed and would not save all of the directory's files and subdirectory contents. Thus, some data files could be lost during a restore operation because the original files would not be contained in the incremental save set.
3. No attempts were made to restore alias or synonym file entries.

These problems have been corrected in this version of BACKUP, as follows:

1. BACKUP now examines the target disk and the save-set contents and more properly determines which save-set entries are to be ignored and which target disk entries are to be deleted. Additionally, if a privilege error is encountered when directories or other files are to be deleted from the target disk, an attempt is made to change the protection of the files so they can be deleted.
2. BACKUP now detects that a directory file has been modified and saves the directory contents and all of its subdirectories' contents. This will allow renamed directories to be properly restored.

Note

Note that the renaming of directories is a procedure that is not recommended by Digital. Also note that changing security information for a directory changes its modification date. Thus, directories might appear to be "renamed" and their contents and subdirectories' contents included in incremental save sets if their file protection or security information is changed. The addition of renamed directory contents might increase the size of some incremental save sets.

In incremental restore operations, BACKUP processes the target disk directory structure by directory levels in alphabetical order. Thus, circumstances may occur that prevent BACKUP from properly restoring an incremental save set to a target disk. An example of this is where the target disk does not have sufficient space to hold newly "renamed" directories and their contents prior to deleting the original directories and their contents on the target disk.

If the incremental restore fails due to insufficient disk space, a possible workaround is to apply that incremental save set a second time (prior to doing anything else). Often, this workaround will complete successfully because the first incremental restore continues and deletes directories and their contents, making more space available on the target disk. Another possible workaround is to selectively restore files from the save set.

3. BACKUP now attempts to restore alias or synonym file entries in incremental restore operations that do not specify multiple processing of alias or synonym file entries (/NOALIAS). In cases where the alias entry cannot be restored properly, an error message is issued indicating the alias file entry, its primary file, and a secondary status indicating the cause of the failure.

   If the /LOG qualifier was specified in the BACKUP command, successful restoration of alias file entries is indicated by a message to the user.

   If there is to be a separate verify pass, (/VERIFY qualifier was specified in the BACKUP command), alias entry restoration is attempted during the verify pass. Otherwise, alias entry restoration is attempted along with the normal file restoration. The reason for this behavior is that an attempt is made to have all primary files restored prior to attempting to restore alias entries that will eventually reference those files.

3.3.2.4 Save Operation Error Message

V6.2

In prior versions of OpenVMS, if a user specified a version number for a disk save set on a save operation and a file with that version already existed, BACKUP would overwrite the file, destroying the contents of the file. This could result in loss of data.

BACKUP now reports the following error message if a file with the specified version already exists:

```
RMS-E-FEX, file already exists, not superseded
```

For example:

```
$ DIRECTORY LOGIN.BCK;*

  Directory WORK$:[BACKUP]

  LOGIN.BCK;1

$ BACKUP LOGIN.COM LOGIN.BCK;1/SAVE
%BACKUP-F-OPENOUT, error opening WORK$:[TEST]LOGIN.BCK;1 as output
 -RMS-E-FEX, file already exists, not superseded
```

3.4 Batch and Print Queue Management

The following notes contain information about managing batch and print queues.

3.4.1 Problems and Restrictions

The following sections describe problems and restrictions pertaining to batch and print queue management. For problems important to programmers, refer to Section 4.1.1.

3.4.1.1 Process Rights Identifier Limits

V6.2

In OpenVMS Version 6.2, processes that have more than 512 rights identifiers cannot perform queuing system operations. When a process with more than 512 identifiers attempts to submit or print jobs, or manage queues, forms, or characteristics, the system returns the following error message:

```
SYSTEM-F-ARBTOOBIG, access rights block too big
```

3.4.1.2 VMScluster Environment

VAX V6.0
Alpha V6.1

The multiple queue manager feature cannot be used until all nodes in a VMScluster environment are running OpenVMS VAX Version 6.0 or later and OpenVMS Alpha Version 6.1 or later.

In addition, once you begin using multiple queue managers, you cannot bring into the cluster a node running an operating system version earlier than those cited above. Doing so can result in unexpected behavior or failures.

3.4.1.3 PRINT/DELETE Command

VAX V5.5
Alpha V1.5

Before OpenVMS VAX Version 5.5 and OpenVMS Alpha Version 1.5, the queue manager allowed users to specify the PRINT/DELETE command for a file residing on a disk that was not mounted clusterwide, as long as the queue specified in the command was assigned to a node with access to the file being printed.

As of OpenVMS VAX Version 5.5 and OpenVMS Alpha Version 1.5, the new clusterwide queue manager process must have access to the file specified with the PRINT/DELETE command. Otherwise, the file is printed but not deleted.

This problem will be addressed in a future release of the operating system. Until then, you can ensure that the PRINT/DELETE command deletes the specified files by mounting the disks on which the files reside clusterwide. To mount a disk clusterwide, use the /CLUSTER qualifier with the MOUNT command.

However, if your operating environment does not allow you to mount a disk clusterwide, you can resolve this problem by running the queue manager process on a node that has access to the disk. You can specify the node on which the queue manager process runs by specifying the /ON=*node-list* qualifier with the START/QUEUE/MANAGER command. For more information on this qualifier, see the *OpenVMS DCL Dictionary*.

3.4.1.4 SUBMIT/DELETE Command

VAX V5.5
Alpha V1.5

The information in Section 3.4.1.3 also applies to the SUBMIT/DELETE command.

3.4.2 Corrections

This section describes corrections to batch and print queue features used by system managers. For batch and print queue corrections related to programming, see Section 4.1.2. For corrections of interest to the general user, see Section 2.1.1.

3.4.2.1 Form Feeding in Print Symbiont

V6.2

In previous releases, the PRTSMB and LATSYM print symbionts would sometimes print an unwanted form feed after a print job. This unwanted behavior occurred under the following conditions:

- The print job ends at the end of a page.
- A file printed with the /PASSALL qualifier ends with a form feed.

The symbiont has been modified to eliminate the unwanted form feeds.

3.4.2.2 Form/Stock Mismatch

V6.2 Beginning with OpenVMS Alpha Version 1.5 and VMS VAX Version 5.5, a print job could unexpectedly be pending due to a stock-type mismatch. The stock on the print job matched the stock on the mounted form; however, a stock-type mismatch was reported by the queue manager.

This problem has been fixed.

3.4.2.3 LAT Initialization Timing

V6.2 In OpenVMS Version 6.1, starting a LAT queue might cause a symbiont process crash, leaving the queue in a stopped state and causing a process dump (SYS$SYSTEM:LATSYM.DMP). Symbiont processes can handle many print queues. This problem would only occur when starting a queue created a new symbiont process.

This problem has been fixed.

3.4.2.4 Memory Leak in Modified Symbiont

V6.2 In previous versions, when a modified print symbiont contained a combination of a user-written input routine and a user-written format routine, memory leaks occurred in the modified print symbiont. The symbiont would eventually abort with the following message:

```
%JBC-E-WRISMBMBX, error writing symbiont mailbox
SYSTEM-W-MBFULL, mailbox is full
-STR-F-INSVIRMEM, insufficient virtual memory
```

This problem has been corrected.

3.4.2.5 STS$V_INHIB_MSG in Return Status

V6.2 In previous versions, when a batch job exited with an error status and the inhibit message flag was set in the return status, the error message was displayed in the job-completion notification message.

This problem has been fixed. Now, if the flag STS$V_INHIB_MSG is set in the return status, the error message is suppressed in the notification message.

3.5 DCL Commands

This section contains information about DCL commands commonly used by system managers.

3.5.1 Changes and Enhancements

This section describes changes and enhancements to DCL commands commonly used by system managers.

3.5.1.1 ANALYZE/ERROR_LOG/SUMMARY Command

V6.2 The /SUMMARY qualifier is no longer supported for the ANALYZE/ERROR_LOG command.

3.5.1.2 MOUNT/UCS_SEQUENCE Command

V6.2 ISO 9660 volumes might contain Supplementary Volume Descriptors (SVDs) that specify a graphic character set. When selected when the volume is mounted, this graphic character set is used as the default character set for displaying the volume's directories and file names.

The /UCS_SEQUENCE=(escape_sequence) qualifier to the MOUNT command supplies the escape sequence to select the coded graphic character set, which is required when mounting an ISO 9660 volume for one of its SVDs.

All ISO 9660 volumes contain a Primary Volume Descriptor (PVD) that uses ASCII (ISO 646-IRV) as the character set. Both ISO 9660 and OpenVMS file-naming conventions use the same subset of ASCII characters when displaying a volume's directories and file names.

3.5.2 Corrections

This section describes corrections to problems associated with DCL commands commonly used by system managers.

3.5.2.1 ANALYZE/DISK_STRUCTURE/REPAIR Command

V6.2

In previous versions of OpenVMS, the following problems existed:

- If you ran ANALYZE/DISK_STRUCTURE/REPAIR on a device that was write-locked, the command sometimes unlocked the device before exiting.
- If you used STOP/IDENTIFICATION=pid to stop a process while it was running ANALYZE/DISK_STRUCTURE/REPAIR on a device, the device remained write-locked.

These problems have now been corrected.

3.5.2.2 SET ACCOUNTING/NEW_FILE Command

V6.2

Previously, when all account events were disabled and there were no accounting files, the SET ACCOUNTING/NEW_FILE command would create two accounting files.

This problem has been corrected.

3.6 DECdtm Services in a DECnet/OSI Network

This section contains notes pertaining to DECdtm services.

3.6.1 Problems and Restrictions

This section describes known problems and restrictions associated with using DECdtm services.

3.6.1.1 Assigning SCSNODE Values

V6.1

Read this section if all of the following apply to you:

- You have a DECnet/OSI network.
- You use DECdtm services.
- Your DECdtm transactions span different VMScluster systems or standalone computers.

SCSNODE is a system parameter that defines the name of a computer. DECdtm transactions can fail if the same SCSNODE value is used for more than one computer.

Make sure the same SCSNODE name is not duplicated on other computers by following these steps:

1. Make a note of the computers that belong to your **transaction group**. A transaction group is a group of computers involved in DECdtm transactions, where:
   - A computer belongs to only one transaction group.
   - Every computer in a VMScluster belongs to the same transaction group.

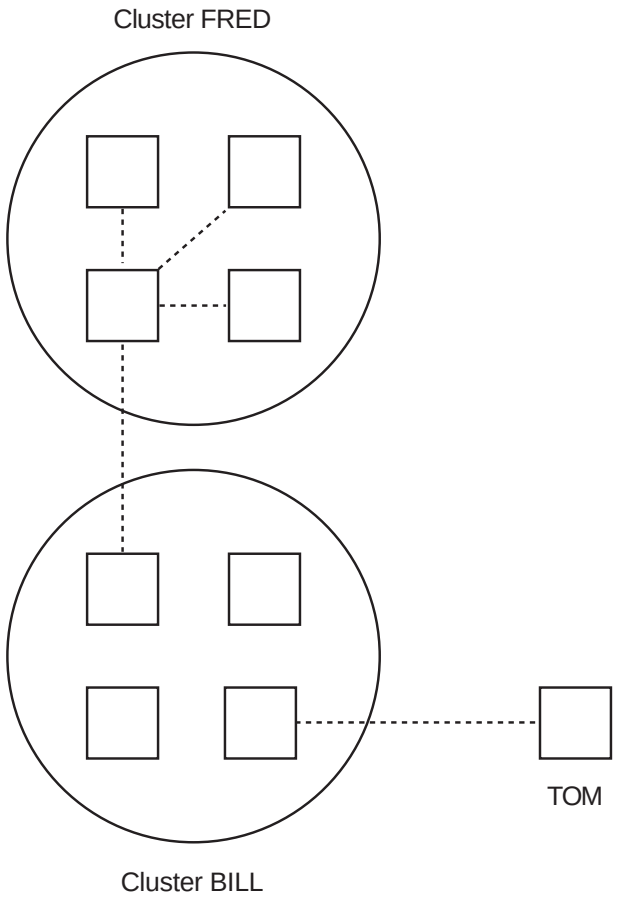
- Computers A and B belong to the same transaction group if any transaction on computer A involves computer B.
- Computers A and C belong to the same transaction group if any transaction on computer A involves computer B, and any transaction on computer B or any node in B's VMScluster involves computer C.

Figure 3–1 shows an example of a transaction group.

2. For each computer in your transaction group, make sure that the SCSNODE value is different from:
   - The SCSNODE values of other computers in the transaction group
   - DECnet synonyms of other computers in the entire network
   - DECnet simple names of other computers on the same local root
3. If the computer is part of a VMScluster, also make sure that the SCSNODE value is different from:
   - DECnet simple names of other computers in the same VMScluster
   - DECnet simple names of computers on the same local root as other VMScluster members

For information on how to find out DECnet synonyms and DECnet simple names, see the *DECnet/OSI DECdns Management* manual. For information on how to find out or change the SCSNODE name, see the *OpenVMS System Manager's Manual*.

Figure 3–1 Transaction Group



Note the following in the example in Figure 3–1:

- Transactions on a computer in cluster FRED involve other computers in cluster FRED and a computer in cluster BILL.
- Transactions on a computer in cluster BILL involve standalone machine TOM.
- No other computers in the network are involved in transactions with computers in clusters FRED or BILL, or with standalone computer TOM.

Therefore, the computers in the transaction group are:

All computers in cluster FRED
All computers in cluster BILL
Computer TOM

3.7 DECevent Fault Management Support (Alpha Only)

V6.2

Alpha

Beginning with OpenVMS Alpha Version 6.2, DECevent replaces the Error Reporting Formatter (ERF) as the bit-to-text translating tool for fault management on OpenVMS Alpha systems. While the ERF is still available on Version 6.2, it will be retired in future releases of the OpenVMS Alpha operating system.

3.7.1 Problems and Restrictions

This section describes problems and restrictions related to using the DECevent fault management tool.

3.7.1.1 Bit-to-Text Translation Support

V6.2

DECevent bit-to-text translation is supported on many products and devices. On other devices, as much translation as possible is performed and all remaining information in the event is dumped in hexadecimal.

Contact your Digital support representative if you have questions about the type of DECevent support currently available for your devices.

3.7.1.2 Ctrl/C

V6.2

After pressing Ctrl/C, you must also press Return. Otherwise, no prompt is returned.

3.7.1.3 Log Files Not Purged Automatically

V6.2

DECevent does not automatically purge log files. Either purge them manually or set version limits on the files and directory. For example:

```
$ SET FILE/VERSION=3 DIA_ACTIVITY.LOG
```

3.7.1.4 Logical File Names

V6.2

DECevent is unable to translate as input any logical defined as a search list of file names. For example:

```
$define event_log disk1:[events]event_log1.sys,disk1:[events]event_log2.sys
$dia/analyze event_log

DECevent T1.0 FT2
_DIAGNOSE-FAT: Analyze - No files found ' event_log '
_DIAGNOSE-FAT: An error occurred while executing a command ruleset
_DIAGNOSE-INF: No Error Messages to send in thread 1
```

3.7.1.5 Page File Quota

V6.2

The possibility exists that DECevent will fail with an access violation if the page file quota is exceeded. If this happens, the process terminates and you are returned to the system prompt. You can then reenter the last failing command.

3.7.1.6 Unrecognized Error Log Messages

V6.2

The DIAGNOSE command does not recognize error log messages logged using the $SNDERR system service. ♦

3.8 DECnet for OpenVMS Alpha (Phase IV)

Alpha

The following sections describe new support and restrictions that apply to DECnet for OpenVMS Alpha (Phase IV). For information about using DECnet/OSI (Phase V), see the DECnet/OSI documentation.

3.8.1 Problems and Restrictions

V1.5 The following restrictions apply to DECnet for OpenVMS Alpha:

- Level 1 routing is available but is supported only on DECnet for OpenVMS Alpha nodes acting as routers for a cluster alias. Routing between multiple circuits or having more than one routing circuit enabled at once is not supported.
- Level 2 routing is not supported on DECnet for OpenVMS Alpha nodes.
- Some line types are unsupported.

  DECnet for OpenVMS Alpha nodes can connect to a DECnet network only via Ethernet lines or FDDI lines. DECnet communication over CI lines is not supported. There also is no support for DDCMP lines.

  Because DDCMP lines are unsupported, the DCL command SET TERMINAL /PROTOCOL=DDCMP /SWITCH=DECNET also is unsupported on Alpha systems. ♦

3.9 DECnet for OpenVMS VAX (Phase IV)

VAX

The following sections describe new support and restrictions that apply to DECnet for OpenVMS VAX (Phase IV). For information about using DECnet/OSI (Phase V), see the DECnet/OSI documentation.

3.9.1 Changes and Enhancements

The following section describes changes to DECnet for OpenVMS VAX (Phase IV).

3.9.1.1 DEFQA Now Can Be Configured with SYS$MANAGER:NETCONFIG.COM

V6.2 The DEC FDDIcontroller/Q–bus Controller (DEFQA) is now supported by DECnet for OpenVMS VAX (Phase IV).

The DEFQA can now be configured using the DECnet for OpenVMS VAX NETCONFIG.COM procedure. The NCP LINE and CIRCUIT name for the DEFQA is FQA-*n*, where *n* is the number representing the controller; for example, line FQA-0 maps to device FQA0 and line FQA-1 maps to device FQB0.

If you previously had the following logical defined in SYS$MANAGER:SYCONFIG.COM in order to use this device on OpenVMS VAX Version 6.1, you must remove the logical from SYCONFIG.COM and deassign it before configuring the device. This logical is no longer necessary to configure the DEFQA.

```
$ DEFINE/SYSTEM FXc0 FQc0:
```

3.9.2 Problems and Restrictions

The following section describes a restriction pertaining to DECnet for OpenVMS VAX (Phase IV).

3.9.2.1 DEFTA Cannot Be Configured with SYS$MANAGER:NETCONFIG.COM

V6.1 DECnet for OpenVMS VAX cannot configure the DEC FDDIcontroller/TURBOchannel Controller (DEFTA).

To use DECnet for OpenVMS with the DEFTA on OpenVMS VAX Versions 6.1 and 6.2, you must define a logical name before invoking the NETCONFIG.COM or STARTNET.COM command procedures. Use the following DCL command to define the required logical name:

```
$ DEFINE/SYSTEM FZc0 FCc0:
```

In this command, *c* represents the controller (for example, FCA0). To make the definition permanent, define this logical name in SYS$MANAGER:SYCONFIG.COM.

The NCP LINE and CIRCUIT name for the DEFTA controller is FZA-*n*, where *n* is the number representing the controller; for example, line FZA-0 maps to device FZA0 and line FZA-1 maps to device FZB0. ♦

3.10 DECnet/OSI for OpenVMS VAX (Phase V)

VAX

The following sections contain notes that apply to DECnet/OSI (Phase V). For complete information about using DECnet/OSI, see the DECnet/OSI documentation.

3.10.1 Changes and Enhancements

The following section describes changes to DECnet/OSI.

3.10.1.1 DEFQA Now Can Be Configured with SYS$MANAGER:NET$CONFIGURE.COM

V6.2

The DEC FDDIcontroller/Q–bus Controller (DEFQA) is now supported by DECnet/OSI for OpenVMS VAX (Phase V).

The DEFQA can be configured by DECnet/OSI for OpenVMS VAX using the NET$CONFIGURE.COM procedure. For information on how to configure devices using DECnet/OSI, refer to the *DECnet/OSI for OpenVMS Installation and Configuration* manual.

If you previously had the following logical defined in SYS$MANAGER:SYCONFIG.COM in order to use this device on OpenVMS VAX Version 6.1, you must remove the logical from SYCONFIG.COM and deassign it before configuring the device. This logical is no longer necessary to configure the DEFQA.

```
$ DEFINE/SYSTEM FXc0 FQc0:
```

3.10.1.2 DEFTA Now Can Be Configured with SYS$MANAGER:NET$CONFIGURE.COM

V6.2

The DEC FDDIcontroller/TURBOchannel Controller (DEFTA) is now supported by DECnet/OSI for OpenVMS VAX (Phase V).

The DEFTA can be configured by DECnet/OSI for OpenVMS VAX using the NET$CONFIGURE.COM procedure. For information on how to configure devices using DECnet/OSI, refer to the *DECnet/OSI for OpenVMS Installation and Configuration* manual.

If you previously had the following logical defined in SYS$MANAGER:SYCONFIG.COM in order to use this device on OpenVMS Version 6.1, you must remove the logical from SYCONFIG.COM and deassign it before configuring the device. This logical is no longer necessary to configure the DEFTA using DECnet/OSI.

```
$ DEFINE/SYSTEM FZc0 FCc0: ♦
```

3.11 DECwindows Motif for OpenVMS

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

3.11.1 Changes and Enhancements

This section includes notes about support for the the DECwindows Motif for OpenVMS layered product.

3.11.1.1 DECwindows Motif Version 1.1 for OpenVMS

V6.2

OpenVMS Version 6.2 is the final version of the OpenVMS operating system to support DECwindows Motif Version 1.1 for OpenVMS. Future versions of OpenVMS will require that DECwindows Motif Version 1.2 or later be installed.

3.11.1.2 NCSA Mosaic Browser: TCP/IP Support Options

V6.2

DECwindows Motif Version 1.2–3 includes the NCSA Mosaic web browser. The *DECwindows Motif Version 1.2–3 for OpenVMS Release Notes* state that the NCSA Mosaic browser requires that the TCP/IP Services for OpenVMS product (UCX) be installed on your system. Note that TCP/IP support can alternatively be provided by PathWay for OpenVMS (Wollongong) or TCPware (Process Software).

3.11.2 Problems and Restrictions

This section describes problems and restrictions associated with the DECwindows Motif for OpenVMS layered product.

3.11.2.1 Console Broadcasts Disabled

V6.2

In Version 1.2–3, console broadcasts are disabled by default by the DECwindows startup procedure on nonworkstation systems as well as on workstation systems. To allow broadcasts to OPA0:, edit the file SYS$MANAGER:DECW$PRIVATE_APPS_SETUP.COM (creating it if it does not exist) and add the following global symbol definition:

```
$ DECW$CONSOLE_SELECTION == "ENABLE"
```

Then, restart DECwindows by entering the following command:

```
$ @SYS$MANAGER:DECW$STARTUP RESTART
```

On workstation systems, Digital recommends that you set DECW$CONSOLE_SELECTION to WINDOW instead of ENABLE. This directs console output to a Console Window application, which is new in DECwindows Motif V1.2–3, instead of to the operator window on the graphics screen.

3.11.2.2 System Files Purged During Startup

V6.2

In Version 1.2–3, the following DECwindows files are purged each time DECwindows Motif starts:

SYS$LIBRARY:DECW$*.EXE
SYS$SYSTEM:DECW$SETSHODIS.EXE

3.11.2.3 Virtual Data Error Message (Alpha Only)

V6.2

Alpha

In Version 1.2–3, the following error message is displayed on nonworkstation OpenVMS Alpha systems during system startup:

```
%SDA-E-NOTINPHYS, 00000024 : virtual data not in physical memory
```

You can ignore this message. ♦

3.11.2.4 World Wide Support Requires Remedial Kit

V6.2

Before you upgrade to OpenVMS Version 6.2 on systems that run language variant versions of DECwindows Motif Version 1.2, you should first install a remedial kit for DECwindows Motif Version 1.2 World Wide Support. This patch is necessary because the internationalization support provided in OpenVMS Version 6.2 is incompatible with the support in DECwindows Motif Version 1.2.

The name of the remedial kit is DWMV_I18N01_012 on VAX and DWMA_ I18N01_012 on Alpha. If you install OpenVMS Version 6.2 from the CD–ROM, the remedial kit will be on the CD–ROM; otherwise you must obtain the kit from your Digital support representative.

Note

Systems running language variant versions of DECwindows Motif Version 1.2 should not be upgraded to DECwindows Motif Version 1.2–3 until a World Wide Support kit becomes available for DECwindows Motif Version 1.2–3.

3.11.2.5 DECwindows Motif Version 1.2 Compatibility Problem

V6.2

DECwindows Motif Version 1.2 issues the following warning message when it encounters compatibility problems with OpenVMS Version 6.2:

```
X Toolkit Warning: locale not supported by C library, locale unchanged
```

In some cases, applications can also terminate with an ACCVIO status.

A new system logical name has been introduced as a temporary workaround to prevent this problem from occurring when running DECwindows Motif Version 1.2–3. The logical name definition is included in VMS$LPBEGIN-050_ STARTUP.COM:

```
$ DEFINE/SYSTEM DECW$USEXLIBXPG4 TRUE
```

Using this system logical name enables applications to work as they have in the past. However, defining the logical name has no effect on DECwindows Motif Version 1.2.

Therefore, if you have upgraded to OpenVMS Version 6.2 and are running DECwindows Motif Version 1.2 and are *not* running a language variant of DECwindows Motif, Digital recommends that you also upgrade to DECwindows Motif Version 1.2–3.

Alternatively, you can work around this problem by applying a remedial kit to DECwindows Motif Version 1.2. Contact your Digital support representative for the following kits that add support for the DECW$USEXLIBXPG4 logical name:

	Kit Number	
Platform	**For Base Product**	**For WorldWide Support**
Alpha	AXPMOTF05_012	DWMA_I18N01_012
VAX	VAXMOTF05_012	DWMV_I18N01_012

3.11.2.6 DECwindows Server Height or Width Exceeding 32767 (VAX Only)

V6.1

VAX

When an X application sends the display server a width or height greater than 32767, the application may terminate with a BadValue error similar to the following:

```
X error event received from server: BadValue (integer parameter out of
range for operation)
  Major opcode of failed request: 61 (X_ClearArea)
  Value in failed request: 0xffff****
  Serial number of failed request: ###
  Current serial number in output stream: ###
```

The following calls can cause this problem:

CopyArea()
CreateWindow ()
PutImage()
GetImage()
CopyPlane()
ClearArea()

This is due to the width and height being defined as a signed word by the display server when it should be defined as an unsigned word (CARD16) that allows for values up to 65536.

To modify the default operation perform the following steps:

1. Set the logical name DECW$CARD16_VALIDATE to TRUE as follows:

```
$DEFINE/TABLE=DECW$SERVER0_TABLE   DECW$CARD16_VALIDATE  TRUE
```

2. Exit the session and log back in.

   Exiting the session causes the display server to reset using the new value of the logical name DECW$CARD16_VALIDATE. The server will now accept values that are greater then 32767 without generating an error.

To make this a permanent change, add the command from Step 1 to the file SYS$MANAGER:DECW$PRIVATE_SERVER_SETUP.COM.

3.11.2.7 SET DISPLAY Used to Create WSA Pseudo Workstation Devices

V6.1

When creating WSA pseudo workstation devices using the SET DISPLAY command, be careful not to create WSA devices that are never destroyed. For example, this DCL command procedure is wrong:

```
$LOOP:
$  SET DISPLAY/CREATE/NODE=remote
$  RUN SYS$SYSTEM:DECW$CLOCK
$  IF $STATUS THEN GOTO DONE
$  WAIT 0:0:5
$  GOTO LOOP
$DONE:
```

If the clock cannot be started for some reason, one WSA device will be created for each failed attempt. These WSA devices will use up non-paged dynamic memory, and eventually the process will exceed its BYTLM quota and enter a resource wait state (if resource waiting is enabled, as it is by default).

A better version of this command procedure is the following:

```
$  SET DISPLAY/CREATE/NODE=remote
$LOOP:
$  RUN SYS$SYSTEM:DECW$CLOCK
$  IF $STATUS THEN GOTO DONE
$  WAIT 0:0:5
$  GOTO LOOP
$DONE:
$  SET DISPLAY/DELETE 'F$TRNLNM("DECW$DISPLAY")'
```

The SET DISPLAY/DELETE command deletes the WSA device that was created at the beginning of the command procedure; the logical name DECW$DISPLAY contains the name of the WSA device that was created.

3.11.2.8 DECwindows Motif Start Session Dialog Box Not Displayed After Rebooting the System (VAX Only)

V6.0

VAX

Occasionally when you reboot a system running DECwindows Motif Version 1.1 software on OpenVMS VAX Version 6.0 or later, the Start Session dialog box does not display.

Usually you can work around the problem by entering the following command from a privileged account:

```
$ @SYS$MANAGER:DECW$STARTUP APPS
```

To resolve the problem, order DECwindows Motif for OpenVMS Version 1.2–3 or the latest available version or contact a Digital support representative to obtain patch kit CSCPAT_1090 for DECwindows Motif Version 1.1 software. ♦

3.12 Help Message Utility (MSGHLP)

This section contains release notes pertaining to the Help Message utility (MSGHLP).

3.12.1 Problems and Restrictions

This section describes problems and restrictions related to using the Help Message utility.

3.12.1.1 User-Supplied Comments Are Not Preserved

VAX V6.0
Alpha V1.0

Currently, user-supplied comments or additions to Digital-supplied .MSGHLP$DATA files will not be preserved through the next upgrade. However, your own .MSGHLP$DATA files are not affected by upgrades.

Note that you can reuse .MSGHLP files to insert your own messages into future Digital-supplied database files. Depending on the content of future databases, you might also be able to reuse some .MSGHLP files to add comments to existing messages.

3.13 InfoServer MADDRIVER

This section contains a note pertaining to the InfoServer MADDRIVER.

3.13.1 Problems and Restrictions

The following section describes an InfoServer MADDRIVER restriction.

3.13.1.1 Magnetic Tapes Are Not Supported (Alpha Only)

V6.2

Alpha

OpenVMS Alpha Version 6.2 does not support magnetic tapes served by the InfoServer. A remedial kit will be supplied in the near future to support this function.

Caution

Until you can install the remedial kit, do *not* attempt to load the existing MADDRIVER. Using it could cause your system to crash. ♦

3.14 INITIALIZE

This section describes notes pertaining to INITIALIZE.

3.14.1 Problems and Restrictions

The following section describes an INITIALIZE problem.

3.14.1.1 Recovery of Badly Initialized Root Directory (Alpha Only)

V6.1

Alpha

In the Software Developer Toolkit (SDK) version of OpenVMS Alpha Version 6.1, INITIALIZE did not initialize the root directory of a volume, 000000.dir correctly. This bad initialization could cause the directory entry for this file to be deleted, and also prevent you from being able to set ACLs on 000000.dir.

To tell whether you have a disk that has suffered from this, use the ANALYZE /DISK_STRUCTURE command on the disk. A new warning message has been added to flag this condition:

```
%ANALDISK-W-BADINITD_MFD, Root directory 000000.DIR;1 file header
incorrectly initialized, RVN 1
```

Use the ANALYZE/DISK_STRUCTURE/REPAIR command to repair this bad initialization.

If the directory entry for 000000.dir has been deleted, then use ANALYZE/DISK_ STRUCTURE/REPAIR to repair this as well. You will receive the following warning from ANALYZE/DISK_STRUCTURE if this directory entry has been deleted:

```
%ANALDISK-W-LOSTHEADER, file (4,4,1) 000000.DIR;1 not found in a directory ♦
```

3.15 Install Utility (INSTALL)

This section contains release notes pertaining to the Install utility (INSTALL).

3.15.1 Changes and Enhancements

This section describes a change to INSTALL.

3.15.1.1 Installing Images

V6.2

The REPLACE option for the OpenVMS Install utility (INSTALL) has been changed to modify the known file database in an atomic fashion.

In the past, REPLACE was equivalent to DELETE followed by ADD. Consequently, there was a short time during which neither the new nor the old image was in the known file database. When activating protected or privileged images, this could result in failed image activations. Also, if the new image could not be installed, it was possible for neither the old nor the new image to be installed after the failure.

These problems are now corrected.

With the change, REPLACE operations for images installed with the /SHARED qualifier might require more global sections or global pages than in the past. Also, the names of global sections have been changed to avoid naming conflicts. The global sections can be displayed with one of the following commands:

```
$ INSTALL LIST /GLOBAL
```

or

```
$ INSTALL LIST image-name /GLOBAL
```

3.16 License Management Facility (LMF)

This section contains information about the License Management Facility (LMF).

3.16.1 Problems and Restrictions

This section describes a restriction pertaining to the License Management Facility (LMF).

3.16.1.1 PAKs on Alpha and VAX Systems

VAX V6.0
Alpha V1.0

Availability Product Authorization Keys (PAKs) are available for OpenVMS Alpha. An OpenVMS Alpha PAK is identified by the keyword ALPHA in the PAK's option field.

PAKs having the ALPHA option will be loaded and used only on Alpha systems. However, they can safely reside in a license database (LDB) shared by both VAX and Alpha systems.

Availability PAKs for VAX systems (availability PAKs without the ALPHA option) will not load on Alpha systems. Only those availability PAKs containing the ALPHA option will load on Alpha systems.

Other PAK types, such as activity (also known as concurrent or n-user) and personal use (identified by the RESERVE_UNITS option), work on both VAX and Alpha systems.

Caution

By default, all Alpha availability PAKs look disabled to a system running OpenVMS VAX Version 6.0 or earlier. Never use the DELETE /STATUS=DISABLED command from a system running OpenVMS VAX

Version 6.0 or earlier on an LDB that contains Alpha PAKs. If you do, all Alpha PAKs will be deleted.

See the *OpenVMS License Management Utility Manual* for more information about using LMF.

3.17 Monitoring Performance History (MPH)

V6.2

Digital invites you to participate in the Digital Product Performance (DPP) program, which monitors and verifies in-field performance of Digital systems at customer and Digital sites. This program provides Digital service, manufacturing, and design engineering organizations with accurate information about the performance of Digital products. Digital's goal is to provide customers with improved reliability on all Digital systems.

To ensure the high quality of its products, Digital has developed a system monitoring tool called Monitoring Performance History (MPH). MPH resides on participants' systems with negligible impact on system performance and no impact on system security.

MPH collects error log entries, crash dump summaries, and configuration information from monitored systems. Every week the information is sent to the DPP group using the selected transport mechanism, which can include DSNLink and Internet mail, among others.

DPP analyzes the collected information and generates reports that are distributed to engineering, manufacturing, and services groups. These groups evaluate this information in an effort to improve system reliability and availability. All collected information is classified as Digital Confidential and is held for internal use only.

The MPH process, with the exception of installation, is fully automated. MPH runs as a background task utilizing negligible system resources. The disk space required for the collected data and the application is approximately 300 blocks per system.

MPH is a voluntary program that requires no special maintenance agreement with Digital.

MPH Kit Location on the Media

The MPH kit and installation guide are in the following OpenVMS Version 6.2 media locations:

- Volume 3 of the magnetic tape media
- Volume 2 of the TK50 media
- Directory [MPH] of the CD–ROM media

Installing MPH

You install MPH by using VMSINSTAL. The installation manual is in the MPH kit and can be extracted in either text form or POSTSCRIPT form.

- To extract the installation manual in text form, enter the following command:

```
$ BACKUP/SELECT=MPH_VMS_INSTALL_GUIDE.TXT MPH_VMS011.A /SAVE [].TXT
```

- To extract the information in POSTSCRIPT form, enter the following command:

```
$ BACKUP/SELECT=MPH_VMS_INSTALL_GUIDE.PS MPH_VMS011.A /SAVE [].PS
```

Stopping MPH

You can stop MPH on your systems at any time by entering this command:

```
$ @SYS$MANAGER:MPH$SHUTDOWN.COM
```

Deinstalling MPH

You can deinstall MPH at any time by entering this command:

```
$ @SYS$MANAGER:MPH$DEINSTAL.COM
```

New Features in Version 1.1 of the OpenVMS MPH Kit

The following new features are included in Version 1.1 of the OpenVMS MPH kit:

- MPH now offers you the opportunity to upgrade your existing installation.
- Error log data can be collected on a weekly or daily basis.
- Customer questions asked during the installation process have been updated to retrieve more accurate information.
- Internet users can now use an Internet gateway, if required.
- MPH now performs space management to ensure it has no impact on your system, regardless of error log size.
- The MPH directory is no longer required to be a top-level directory.
- MPH invokes the Crash Log tool (CRL) if the Crash Log Utility Extractor (CLUE) is not running on the system.
- Several new safeguards and warning messages have been introduced; for example:
  - The system manager is notified if there are problems with DSNLink.
  - The system manager is notified if the initial error log is too large or if daily changes to the error log are large.
- After installation on a cluster, monitored nodes can be added or deleted by executing MPH$AREA:MPH$EDIT_NODES.COM.

For more information...

For additional information about MPH or the DPP program, send mail to:

```
mph_admin@dppsys.enet.dec.com
```

Include the keyword INFO anywhere in the subject header of your message to expedite routing of your request.

Digital looks forward to your participation in this mutually beneficial program. Thank you for your cooperation.

3.18 Monitor Utility (MONITOR)

The following notes pertain to the Monitor utility (MONITOR).

3.18.1 Changes and Enhancements

This section describes changes to the Monitor utility.

3.18.1.1 Disk Limit Increased

V6.2

The limit on the number of disks that can be monitored on OpenVMS Version 6.2 has been raised from 799 to 909 for record output and from 799 to 1817 for display and summary outputs. If you are monitoring a remote node running OpenVMS Version 6.2 from a system running an earlier version of OpenVMS, the old limit of 799 applies.

3.18.1.2 Display Header Changed From "VAX/VMS" To "OpenVMS" (VAX Only)

V6.2

VAX

The text in the first heading line for MONITOR display and summary output has been changed slightly. The heading line "VAX/VMS Monitor Utility" used in previous VAX versions now reads "OpenVMS Monitor Utility" to match OpenVMS Alpha. This change is a reflection of the functional equivalence of MONITOR on both platforms.

Any software that parses this line of the display or summary output of the MONITOR utility should be changed accordingly. This change does not affect MONITOR's binary recording file output. ♦

3.19 Mount Utility (MOUNT)

The following notes pertain to the Mount utility (MOUNT).

3.19.1 Problems and Restrictions

This note describes a MOUNT restriction.

3.19.1.1 ISO 9660 Support in Mixed-Version VMScluster Systems

V6.1

On OpenVMS systems, do not specify the /CLUSTER qualifier when mounting ISO 9660 formatted CD–ROMs in a VMScluster system with nodes that are running versions of OpenVMS VAX prior to Version 6.0. If you attempt to mount an ISO 9660 CD–ROM on an OpenVMS node without ISO 9660 support, which is likely in a mixed-version environment, the operation will fail. However, the failure will take an excessive amount of time to complete, due to the slow access time for the CD–ROM media.

3.19.2 Corrections

This section describes software corrections to MOUNT.

3.19.2.1 Noncompliant ISO 9660 CD–ROMs and MOUNT (VAX Only)

V6.2

VAX

Previously, on OpenVMS VAX systems, support was not provided for mounting noncompliant ISO 9660 CD–ROMs.

If users had noncompliant media and were running applications in a mixed-architecture VMScluster system (OpenVMS Alpha Version 6.1 with either OpenVMS VAX Version 6.0 or OpenVMS VAX Version 6.1), they could experience problems. This was an issue for PATHWORKS users who were using noncompliant ISO 9660 CD–ROMs in a mixed-architecture VMScluster system.

This problem has been corrected. ♦

3.20 Nonpaged Pool

This section contains release notes pertaining to nonpaged pool.

3.20.1 Problems and Restrictions

This section describes software problems and restrictions related to nonpaged pool.

3.20.1.1 Prepopulation of Nonpaged Pool (VAX Only)

V6.2

VAX

Feedback from some sites suggests that prepopulation of the nonpaged pool lookaside lists can cause problems in some environments. For OpenVMS VAX Version 6.2 the SYSGEN parameter VMS7 is temporarily being used as a switch to turn off prepopulation.

If you experience any problems that you believe are related to nonpaged pool prepopulation, set VMS7 to 1. ♦

3.20.2 Corrections

This section describes software corrections relating to nonpaged pool.

3.20.2.1 Nonpaged Pool Deallocation

V6.2

In previous versions of OpenVMS, a problem existed that reduced the amount of free nonpaged pool and that occasionally caused the system to hang or crash.

If you exceed the version limit for a file, and the oldest version of the file is a file alias, then the file system does the following:

1. Creates a directory entry for the new file version
2. Removes the directory entry for the old file alias

Each time this happens, a small amount of nonpaged pool is allocated to the file system.

Previously, the file system failed to deallocate the nonpaged pool at the end of the operation. If the file system performed this operation many times, the system ran out of nonpaged pool, causing the system to hang or crash.

This problem has been corrected.

3.21 OPCOM

The release notes in this section pertain to OPCOM, the operator communication manager.

3.21.1 Problems and Restrictions

The release notes in this section describe restrictions you need to be aware of when using OPCOM.

3.21.1.1 OPERATOR.LOG File

V6.2

The size of and access to the OPERATOR.LOG file (or to the file pointed to by the logical OPC$LOGFILE_NAME) is limited by the size and access of the disk device on which it resides. If the disk device does not have enough room to write to the log file or if access to the device in any other way is restricted, records might be missing from the log file.

3.22 POLYCENTER Software Installation Utility

The notes in this section pertain to the POLYCENTER Software Installation utility. Also see Section 4.21 for notes about this utility that are of interest to programmers.

3.22.1 Problems and Restrictions

Notes in this section pertain to problems and restrictions with using the POLYCENTER Software Installation utility to install, remove, and reconfigure software products. Problems and restrictions of interest to programmers are described in Section 4.21.1.

3.22.1.1 Dependency Information Is Lost On Product Upgrade

V6.2

When a software product is upgraded to a new version, information about other products that are dependent on it is incorrectly removed from the POLYCENTER Software Installation utility database. Consider this example:

1. Product A Version 2.1 is installed.
2. Product B Version 1.0, which requires Version 2.0 or higher of product A, is installed.
3. Product A is upgraded to Version 2.2, causing the dependency of Product B on Product A to be erroneously erased from the database.

To restore the dependency information, you must reconfigure all products that are dependent on the upgraded product. Reconfiguration of an installed product requires that its software product kit be accessible to the utility.

For example, to restore the dependency information for Product B, shown in the above sequence, place Product B's software kit online and reconfigure Product B as shown in the following code sequence:

```
$ PRODUCT SHOW PRODUCT A/FULL     ! A is referenced by B
$ PRODUCT INSTALL A/VERSION=2.2/SOURCE=disk:[directory]
$ PRODUCT SHOW PRODUCT A/FULL     ! no dependency is shown
$ PRODUCT RECONFIGURE B/SOURCE=disk:[directory]
$ PRODUCT SHOW PRODUCT A/FULL     ! A is referenced by B
```

Digital expects to correct this problem in a future release.

3.22.1.2 PRODUCT Command

There are several problems with the DCL command PRODUCT:

V6.2

- If the /SOURCE qualifier is not specified and the logical name PCSI$SOURCE is not defined, the PRODUCT command does not use the default device and directory as documented; instead, a message reports an error in the device name.

  To avoid this problem, use the /SOURCE qualifier or define the logical name PCSI$SOURCE as the source device and directory for these PRODUCT subcommands: CONFIGURE, COPY, EXTRACT RELEASE_NOTES, FIND, INSTALL, RECONFIGURE, and REGISTER PRODUCT.

- Commands such as PRODUCT FIND and PRODUCT SHOW that can display more than a screen of data do not support the /OUTPUT qualifier. As a result, displayed text can scroll off the screen.

V6.1

- The /DIRECTORY and /DEVICE qualifiers do not work correctly on the PRODUCT SHOW OBJECT command. The utility either ignores them or provides output that does not match the qualifiers.
- The PRODUCT command can cause indexed and relative files to be truncated during a package or install operation. To determine if a particular file is susceptible to being truncated, use the DIRECTORY/SIZE=ALL command to examine the file. If the two sizes are the same, your file will not exhibit this problem. If the two sizes are not the same, perform the following SET FILE command prior to a package or install:

  ```
  $ SET FILE/END_OF_FILE file-in-question.dat
  ```

  This command should be issued only for an indexed or relative file, since the end-of-file (EOF) attribute is meaningful for sequential files.

Digital expects to correct all of these problems in a future release.

3.22.1.3 Deleting Directories Created by the POLYCENTER Software Installation Utility

V6.1

If a directory is created as part of a product installed using the POLYCENTER Software Installation utility and that directory is deleted manually, a reinstallation of the same product will fail. The POLYCENTER Software Installation utility determines from its database that the directory should be there, so it does not issue the command to create it. When the utility tries to reinstall the file, it fails.

Digital recommends that you not manually delete objects that were created by the POLYCENTER Software Installation utility.

3.22.1.4 Inaccurate Disk Space Reporting

V6.1

The POLYCENTER Software Installation utility estimates the amount of disk space required for an installation. Sometimes this number is inaccurate, and an installation may fail due to lack of disk space even though the utility reports that there is enough disk space.

The utility also fails to check for enabled disk quotas.

Digital expects to correct these problems in a future release.

3.22.1.5 Installing an Earlier Version of an Installed Product

V6.1

If you attempt to install an earlier version (one with a lower version number) of an installed product, the utility displays a sequence of questions. The questions do not display the correct defaults; if you take the defaults, the utility attempts to install both versions. This causes an error similar to the following:

```
%PCSI-E-FAILCONF, failed to resolve conflicting requirements for file
[000000]DEC-VAXVMS-XYZ-V0105--1.PCSI$DESCRIPTION
Terminating is strongly recommended.  Do you want to terminate? [YES]
```

The following example shows how to answer the questions to install the version you want:

1. The system displays the following message:

   ```
   %PCSI-E-CONREMHV, confirm removal of higher version of DEC AXPVMS XYZ V2.0
   Do you want to take this action? [NO]
   ```

   If you want the lower version, answer YES. If you want the higher version, answer NO.

2. The system displays the following message:

```
Do you want to continue? [YES]
```

If you want the lower version, answer YES. If you want the higher version, answer NO.

3.22.1.6 Product Removal Restrictions

V6.1

Removing a product using the POLYCENTER Software Installation utility results in the removal of accounts created for that product. This happens regardless of whether the SYSUAF.DAT file is shared by another system disk.

The same problem exists with rights identifiers and the file RIGHTSLIST.DAT.

Digital expects to correct these problems in a future release.

3.22.2 Documentation Changes and Corrections

This section describes changes to documentation of the POLYCENTER Software Installation utility. Other documentation corrections of interest to programmers are described in Section 4.21.2.

3.22.2.1 *POLYCENTER Software Installation Utility User's Guide*

V6.2

Information formerly included in the *POLYCENTER Software Installation Utility User's Guide* has been revised and incorporated into the *OpenVMS System Manager's Manual: Essentials*.

3.23 POLYCENTER Striping (VAX Only)

VAX

This section contains release notes pertaining to the POLYCENTER Striping product.

3.23.1 Changes and Enhancements

This section describes the replacement for POLYCENTER Striping.

3.23.1.1 StorageWorks RAID Software for OpenVMS Replaces POLYCENTER Striping

V6.2

With OpenVMS Version 6.2, POLYCENTER VAX Striping is no longer supported. The supported replacement product is StorageWorks RAID Software for OpenVMS. The Striping functionality in StorageWorks RAID Software for OpenVMS is available to POLYCENTER VAX Striping customers at no charge. Please contact your Digital support representative for details of the migration plan. ♦

3.24 RMS Journaling

The following notes pertain to RMS Journaling for OpenVMS.

3.24.1 Problems and Restrictions

The following notes describe restrictions to RMS Journaling for OpenVMS.

3.24.1.1 Remote Access of Recovery Unit Journaled Files

V6.1

Nodes that host recovery unit journaled files that are to be accessed remotely from other nodes in the network must define SYS$NODE to be a Phase IV style node name. The node name specified by SYS$NODE must be known to any remote node attempting to access the recovery unit journaled files on the host node, and must be sufficiently unique for the remote node to use this node name to establish a DECnet connection to the host node.

3.24.1.2 VFC Format Sequential Files

VAX V5.0
Alpha V1.0

You cannot update variable fixed-length control (VFC) sequential files when using before-image or recovery unit journaling. The VFC sequential file format is indicated by the symbolic value FAB$C_VFC in the FAB$B_RFM field of the FAB.

Digital expects to remove this restriction in a future release of RMS Journaling for OpenVMS.

3.25 Security

This section contains release notes pertaining to system security.

3.25.1 Changes and Enhancements

This section contains changes and enhancements to system security.

3.25.1.1 Auditing Events (VAX Only)

V6.2

VAX

In previous versions of OpenVMS VAX, ill-formed audit requests were recorded through the FATAL BUGCHECK mechanism; that is, the system crashed when the audit server detected an audit request that was not specified correctly.

A second method of handling ill-formed audit requests is now available on OpenVMS VAX. (This method was previously available on OpenVMS Alpha.) Instead of using the FATAL BUGCHECK mechanism, messages that describe what happened are recorded in the audit log file.

Both OpenVMS VAX and OpenVMS Alpha provide the auditing of ill-formed audit requests by default. ♦

3.25.2 Corrections

This section describes software corrections pertaining to system security.

3.25.2.1 Intrusion Services

V6.2

The checking of privileges for intrusion services has been changed. Previously, the wrong privilege bit (SYSPRV) was being checked on intrusion calls.

This problem has been corrected. The correct privilege bit (SECURITY) is now checked.

3.26 Shared Linkage Sections

This section contains notes pertaining to shared linkage sections.

3.26.1 Problems and Restrictions (Alpha Only)

V6.1

Alpha

If you want to use an alternate version of any library installed with shareable linkage, it is essential to use alternate (noninstalled) versions of all the libraries that call that library. The libraries that can be installed with shared linkage are LIBOTS, LIBRTL, CMA$TIS_SHR, DPML$SHR, and DECC$SHR.

The dependencies are in the order listed.

For example, if you issue the command:

```
$ DEFINE DPML$SHR SYS$LIBRARY:DPML$SHR.EXE;
```

then you must also issue the following command:

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

Failure to redefine all calling libraries may result in access violations. ♦

3.27 SORT32 Work Files

This section contains release notes pertaining to SORT32 work files.

3.27.1 Changes and Enhancements

This section describes SORT32 changes and enhancements.

3.27.1.1 Files Now Have Directory Entries

V6.2

Starting with OpenVMS Version 6.1, SORT32 work files were implemented as RMS temporary files to fix a problem wherein SORT32 work files remained if a sort was terminated abnormally. RMS temporary files do not have directory entries and are automatically deleted by RMS upon file closure. Thus, if the SORT32 image was terminated abnormally, the work files were still deleted.

Unfortunately, this implementation caused a system management restriction in that, without a directory entry, no access control information is available for the work files. Therefore, work files could be created only in a directory owned by the SORT32 user.

Typically, for large sorts, the system manager would designate a large shared temporary disk area using access control mechanisms so that the general user would require only small disk quotas. The large work files, which are deleted at the end of the sort, would be placed in the temporary area. Without a directory entry, these access control mechanisms cannot be used and the work file cannot be created in the temporary area.

Work files are now created as regular files with directory entries. Reverting to the previous behavior and allowing the management freedom for work file placement was considered more important than the occasional work file that was not deleted under abnormal conditions. Users can manually delete any work files left after a sort image terminates abnormally.

Infrequently, SORT32 might not be able to find the designated directory for the work file due to some unforeseen reason, such as a user error or hardware malfunction. In this case, SORT32 will attempt to create the work file as an RMS temporary file, possibly allowing the sort to complete successfully. This is a convenience feature and was retained for compatibility with previous versions of SORT32. Alternatively, the creation of SORT32 work files as RMS temporary files can be forced by specifying a nonexistent directory for the location of the work files.

3.28 StorageWorks RAID Array 110 Subsystem (VAX Only)

VAX

This section contains notes pertaining to the StorageWorks RAID Array 110 Subsystem.

3.28.1 Problems and Restrictions

This section describes known problems and restrictions with the StorageWorks RAID Array 110 Subsystem.

3.28.1.1 SHOW DEVICE Does Not Display Capacity

V6.1

On some VAX systems, the capacity of the StorageWorks RAID Array 110 Subsystem is not displayed after you enter the following console command:

```
>>> SHOW DEVICE
```

Instead, the capacity displays as "...". (This is due to the current settings for spin up time in the EEPROM of the StorageWorks RAID Array 110 Subsystem.)

If you reenter the SHOW DEVICE command, the correct capacity will be displayed. This will be corrected in an update to the DEC RAID OpenVMS VAX Utility Kit.

3.28.1.2 TURBOchannel Devices: Booting Not Supported

V6.1

VAX workstations running the OpenVMS VAX operating system do not provide support for system booting from a TURBOchannel device. You cannot, therefore, boot from a StorageWorks RAID Array 110 Subsystem connected to a TURBOchannel-SCSI adapter. ♦

3.29 System Dump Analyzer (SDA)

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

3.29.1 Problems and Restrictions

This section describes known System Dump Analyzer problems and restrictions.

3.29.1.1 EXAMINE Command (VAX Only)

VAX V6.0

VAX

If you make a mistake specifying a virtual address for the EXAMINE command and you are examining global page table entries, your system may crash with a bugcheck. This occurs rarely and only when you use ANALYZE/SYSTEM.

3.30 System Management Utility (SYSMAN)

This section contains release notes pertaining to the OpenVMS System Management utility (SYSMAN).

3.30.1 Changes and Enhancements

This section describes changes and enhancements to the OpenVMS System Management utility (SYSMAN).

3.30.1.1 ALF Command

V6.2

OpenVMS Version 6.2 includes changes to the following SYSMAN ALF commands:

- ADD
- REMOVE
- SHOW

The following are explanations of these changes:

ALF ADD

When you create ALF records for proxy accounts, the <device-name> parameter can be as long as 63 characters. For example:

```
SYSMAN> ALF ADD VMS:.ZKO.VMSORG.SYSMAN.CLIENT1::SYSTEM FOOBAR
```

In this command, VMS:.ZKO.VMSORG.SYSMAN.CLIENT1::SYSTEM is the value for the <device-name> command parameter.

ALF REMOVE

Due to changes in the SHOW command, you can now use REMOVE commands to remove more than one ALF record at a time. For example:

```
SYSMAN> ALF REMOVE TTA*
```

This command removes all matching records that start with the string TTA, but it does not remove any records that start with the string <nodename>$TTA, where <nodename> is the system's SCSNODE name. To remove those records, you must use one of the following commands:

```
SYSMAN> ALF REMOVE <nodename>$TTA*

SYSMAN> ALF REMOVE *TTA*
```

Note that the latter command removes all matching records that contain the string TTA in the device name field. Similarly, the following command removes all matching records that have device names ending with the string TTA:

```
SYSMAN> ALF REMOVE *TTA
```

If you omit either of the wildcard characters and enter a REMOVE command, SYSMAN attempts to match the given device name exactly. If more than one record matches the given criteria, SYSMAN returns an error message. For example, the following command causes SYSMAN to match any record starting with <nodename>$TTA:

```
SYSMAN> ALF REMOVE TTA
```

This command produces the following error message:

```
%SYSMAN-E-ALFWILCRDREQ, more than one record might match - Wildcard or
unit number of device required.
```

Note

This message is new for OpenVMS Version 6.2. Digital recommends that you use caution when issuing REMOVE commands from Version 6.1 or lower SYSMAN clients to Version 6.2 or higher systems.

For example, the following command issued from a system running OpenVMS Version 6.1 or lower to a system running OpenVMS Version 6.2 produces no error messages and deletes all records that match FOOBAR$TTA:

```
SYSMAN> SET ENVIRONMENT/NODE=FOOBAR     ! FOOBAR runs OpenVMS Version 6.2
%SYSMAN-I-ENV, current command environment:
        Individual nodes: FOOBAR
        Username SYSTEM will be used on nonlocal nodes
SYSMAN> ALF REMOVE TTA                  ! Does not produce an error message
SYSMAN>
```

Note that if the same command is issued from a system running OpenVMS Version 6.1 or lower to another system running OpenVMS Version 6.1 or lower, it produces the following error message:

```
%SYSMAN-I-NODERR,  error returned from node FOO
-SMI-E-ALFNOMATCH, no records matched search criteria
```

This is due to incorrect processing of wildcards prior to OpenVMS Version 6.2.

ALF SHOW

The SHOW command now works as documented whether or not you use wildcards. However, there are certain restrictions in wildcard matching of ALF records. For example, the following command displays only those records that start with the string TTA:

```
SYSMAN> ALF SHOW TTA*
```

However, the following command displays only those records that start with the string <nodename>$TTA:

```
SYSMAN> ALF SHOW TTA
```

Similarly, the following command displays records that have device names ending with TTA:

```
SYSMAN> ALF SHOW *TTA
```

If you want to display all records that contain the string TTA, you must enter the following command:

```
SYSMAN> ALF SHOW *TTA*
```

3.30.1.2 Rights Identifiers

V6.2

In OpenVMS Version 6.0, a problem occurred for SYSMAN users having large numbers of rights identifiers associated with their accounts, causing the SMIserver to fail under certain conditions.

In OpenVMS Version 6.1, a temporary restriction was implemented to limit the number of rights identifiers that SYSMAN users could hold. SYSMAN users were limited to 60 rights identifiers per account. This limitation was a total of all process, system, image, and extended rights. If a SYSMAN user running with more that 60 total rights identifiers attempted to invoke a SYSMAN command, the following message was returned:

```
SMI-E-RIGHTSLIM, Temporary restriction:  rights limit exceeded
```

In OpenVMS Version 6.2, this limitation has been modified. The new upper limit for identifiers that a SYSMAN user who is attempting a remote connection can hold is 125.

If a SYSMAN user running with more than 125 total rights attempts to issue a SYSMAN command to a remote node within a cluster, the following error message is displayed:

```
SMI-E-RIGHTSLIM, Rights limit exceeded.
```

Note that this rights limitation includes a minimum of 3 identifiers that are granted during login when the process rights list is created:

- A UIC identifier
- A system identifier
- Depending upon the environment in which the process is operating, at least one environmental identifier

Users who want to run SYSMAN must have either:

- A separate account with no more than 125 rights
- Enough identifiers removed from their current account so that the total number of rights falls within the appropriate range

A system manager usually performs account modification or creation.

3.30.1.3 SHUTDOWN NODE Command Qualifiers Have Changed

V6.2

The former /INVOKE_SHUTDOWN and /NOINVOKE_SHUTDOWN qualifiers have been changed to /INVOKE_SYSHUTDOWN and /NOINVOKE_ SYSHUTDOWN.

The qualifier /INVOKE_SYSHUTDOWN is the new default. Previously, /NOINVOKE_SHUTDOWN was the default.

3.30.2 Problems and Restrictions

The following sections describe known problems and restrictions that apply to the System Management utility (SYSMAN).

3.30.2.1 /CLUSTER Qualifier

VAX V6.0
Alpha V1.5

When the system management environment is set clusterwide with the /CLUSTER qualifier, entering the following SYSMAN command sometimes causes a hang:

```
SYSMAN> DO SHOW DEVICE device-name
```

This hang is caused by a communication failure between the process spawned by the DO command and the remote SMISERVER process on one of the Alpha nodes in the cluster.

To work around this problem, press Ctrl/C to return to the SYSMAN> prompt. Delete the remote SMISERVER process on the node that did not respond. You can then restart the SMISERVER process with the following command:

```
$ @SYS$SYSTEM:STARTUP SMISERVER
```

If the DO SHOW DEVICE *device-name* command hangs on the next attempt, press Ctrl/C to return to the SYSMAN> prompt. Delete and then restart the SMISERVER process on the unresponsive node. Log in to the node that stopped responding, and enter the DO SHOW DEVICE *device-name* command.

Digital expects to fix this problem in a future release of the OpenVMS Alpha operating system.

3.30.2.2 Clusterwide DISKQUOTA Commands (Alpha Only)

V1.5

Alpha

Normally, SYSMAN DISKQUOTA commands for disks that are mounted clusterwide can be performed with the environment set to a single node. The clusterwide operation is done at the XQP level.

If the environment is set clusterwide using the /CLUSTER qualifier, the following message is displayed, and the operation still finishes correctly clusterwide:

```
%SMI-S-DQCLUS, device is mounted clusterwide, CLUSTER environment ignored
```

However, until further notice, when you use the following DISKQUOTA commands for clusterwide operation, you must first set the environment clusterwide by using the /CLUSTER qualifier:

- DISKQUOTA ENABLE/DEV=*device*
- DISKQUOTA DISABLE/DEV=*device*

If these commands are performed with the environment set to a single node, the disk quota for the clusterwide-mounted device is enabled or disabled for that node only.

The following example shows the correct usage and behavior of the DISKQUOTA ENABLE and DISKQUOTA DISABLE commands:

```
SYSMAN> SET ENV/CLUSTER
%SYSMAN-I-ENV, current command environment:
        Clusterwide on local cluster
        Username TESTUSER     will be used on nonlocal nodes

SYSMAN> DO SHOW DEV TEST01$DKA100:
%SYSMAN-I-OUTPUT, command execution on node TEST02
Device                  Device           Error    Volume         Free  Trans Mnt
 Name                   Status           Count     Label        Blocks Count Cnt
TEST01$DKA100:          Mounted              0  TESTVOL         832395     1   4
%SYSMAN-I-OUTPUT, command execution on node TEST03
Device                  Device           Error    Volume         Free  Trans Mnt
 Name                   Status           Count     Label        Blocks Count Cnt
TEST01$DKA100:          Mounted              0  TESTVOL         832395     1   4
%SYSMAN-I-OUTPUT, command execution on node TEST04
Device                  Device           Error    Volume         Free  Trans Mnt
 Name                   Status           Count     Label        Blocks Count Cnt
TEST01$DKA100:          Mounted              0  TESTVOL         832395     1   4
%SYSMAN-I-OUTPUT, command execution on node TEST01
Device                  Device           Error    Volume         Free  Trans Mnt
 Name                   Status           Count     Label        Blocks Count Cnt
TEST01$DKA100:          Mounted              0  TESTVOL         832395     1   4
SYSMAN> DISKQUOTA SHOW */DEV=TEST01$DKA100:
%SYSMAN-I-NODERR, error returned from node TEST02
-SYSTEM-F-QFNOTACT, disk quotas not enabled on this volume
%SYSMAN-I-NODERR, error returned from node TEST03
-SYSTEM-F-QFNOTACT, disk quotas not enabled on this volume
%SYSMAN-I-NODERR, error returned from node TEST04
-SYSTEM-F-QFNOTACT, disk quotas not enabled on this volume
%SYSMAN-I-NODERR, error returned from node TEST01
-SYSTEM-F-QFNOTACT, disk quotas not enabled on this volume
SYSMAN> DISKQUOTA ENABLE/DEV=TEST01$DKA100:
SYSMAN> DISKQUOTA SHOW */DEV=TEST01$DKA100:
%SMI-S-DQCLUS, device is mounted clusterwide, CLUSTER environment ignored
%SYSMAN-I-QUOTA, disk quota statistics on device TEST01$DKA100: --
Node TEST02
     UIC                Usage        Permanent Quota   Overdraft Limit
[0,0]                   0            1000              100
[VMS,TESTUSER]          25           1000              100

SYSMAN> DISKQUOTA DISABLE/DEV=TEST01$DKA100:
SYSMAN> DISKQUOTA SHOW */DEV=TEST01$DKA100:
%SYSMAN-I-NODERR, error returned from node TEST02
-SYSTEM-F-QFNOTACT, disk quotas not enabled on this volume
%SYSMAN-I-NODERR, error returned from node TEST03
-SYSTEM-F-QFNOTACT, disk quotas not enabled on this volume
%SYSMAN-I-NODERR, error returned from node TEST04
-SYSTEM-F-QFNOTACT, disk quotas not enabled on this volume
%SYSMAN-I-NODERR, error returned from node TEST01
-SYSTEM-F-QFNOTACT, disk quotas not enabled on this volume
SYSMAN> ♦
```

3.31 System Time

This section contains notes pertaining to setting the system time.

3.31.1 Changes and Enhancements

This note describes a change to the Watch Chip.

3.31.1.1 Watch Chip — Change in Time Range (Alpha Only)

V1.0

Alpha

Alpha systems maintain their system time during power failures and system down time with a watch chip (BBW). This chip replaces the time-of-day register (TODR) used on VAX systems. The BBW chip allows a range of only one century, placing a greater constraint on the dates that can be accepted by the $SETIME service and SET TIME DCL commands. What used to be a wider date range on VAX systems is now limited to the century between 1957 and 2056.

In addition, a set of sanity checks has been added to the system boot routines to validate the format of the BBW and the values put into it by previous system boots. These checks recognize out-of-bounds values. When the time is known to be earlier than the last time modification or greater than 5 years in the future, you are prompted to enter the time at the console prompt. Such situations may occur when an Alpha computer, after running another operating system such as OSF/1, is rebooted using the OpenVMS Alpha operating system. ♦

3.31.2 Problems and Restrictions

This section describes known problems and restrictions related to setting the system time.

3.31.2.1 Time of Day Register (TODR): Resetting System Time After January 1st (VAX Only)

V6.1

VAX

The following restriction applies to all past and present versions of OpenVMS VAX.

The Time of Day Register (TODR), which the system uses to maintain system time, has a limit of approximately 15 months. If you do not reset the system time after January 1st, the following problems might occur:

- The first time in a new year that you reboot a VAXcluster system or a node in the system, one or more nodes display any of the following:
  - A system time that is a year in the past.
  - A system time that is a year in the future, which might cause passwords to expire and other difficulties.
  - A system time that is correct, but a SHOW SYSTEM command indicates that the system has been up since sometime in the 1800s.
- Even if you have corrected an incorrect system time when you boot the system, one or more of the following problems might remain:
  - A SHOW SYSTEM command displays an incorrect up time such as a date in the 1800s.
  - The error log report (ERRLOG) shows errors for a year in the future.
  - Batch jobs are waiting for a year in the future.
  - Files have a creation or modification date in the future.

OpenVMS maintains system time across reboots using the TODR register in the OpenVMS VAX system CPU. Because the TODR has a range of approximately 15 months, the system actually maintains system time by combining the TODR value with a base time recorded in the base system image (SYS$LOADABLE_IMAGES:SYS.EXE). The base time is defined as follows:

```
01-JAN-CURRENT_YEAR 00:00:00.00
```

Because all TODRs ordinarily have the same base, multiple CPUs can boot off the same system disk, and you can use multiple system disks on one CPU; in either case, the system sets the time correctly.

When a SET TIME command is issued (with or without specifying a time), OpenVMS does the following:

1. Writes the current time to the system image file.
2. Resets the TODR as an offset within the current year.

In a VAXcluster system, the TODR is reset *only on the node on which the SET TIME command was issued*. However, multiple systems might share the system image. This does not normally cause a problem, except after the first day of a new year.

Note

The system issues the SET TIME command when it boots and as a part of the normal SHUTDOWN command procedure.

By December, each node has a very large offset stored in the TODR (from the base time of 1-JAN of that year). When the time advances to a new year, the system image still has the old year and the TODR values are still large.

After January 1st, if a SET TIME command is issued on any node (or any node is shut down using SHUTDOWN.COM), the following happens:

1. The new year becomes the base year.
2. The system resets the TODR on that node.
3. The other nodes still have a large value in the TODR.

After these three events occur, if a node that has a large TODR crashes and rejoins the cluster, its system time is initially in the next year (applying the large TODR to the new year). This system time is recorded as the system's boot time. When the node joins the cluster, its time is set to the correct value, but the boot time remains one year in the future. Certain forms of the SHOW SYSTEM command compare current time to boot time; in this circumstance, SHOW SYSTEM displays incorrect values.

If a system disk is used at different times by different, unclustered CPUs or if different system disks are used at different times on the same CPU, the system might incorrectly set the time to a year in the future or a year in the past, depending on how the CPU's TODR and the value recorded on the system disk become unsynchronized:

- Sharing a system disk across multiple CPUs pushes the time into the future.
- Using multiple disks on one CPU pushes the time into the past.

Action:

Between January 1 and April 1, reset the system time. Use either of the following methods, depending on whether you have a nonclustered node or a node in a VAXcluster system:

Type of Node	Action
Nonclustered node	Use the SET TIME command; for example: `$ SET TIME 05-JUN-1994:12:00:00` [1]

Node in a VAXcluster system	Use the following SYSMAN commands to reset the time on all the nodes in a VAXcluster system; for example:
	`$ RUN SYS$SYSTEM:SYSMAN`
	`SYSMAN> SET ENVIRONMENT/CLUSTER`
	`SYSMAN> SET PROFILE/PRIVILEGE=(LOG_IO,SYSLCK)`
	`SYSMAN> CONFIGURATION SET TIME 05-JUN-1994:12:00:00` [1]
	`SYSMAN> EXIT`

[1]You do not need to specify a time. If you do not specify a time, SET TIME updates the system time using the time in the TODR.

Either method shown in the table resets the TODR and the base time in the system image with the values for the new year.

Note

If you are running the Digital Distributed Time Service (DECdts) on your system, you must use it to set the time. ♦

3.32 Terminal Fallback Facility (TFF) (Alpha Only)

V1.0

Alpha

The Terminal Fallback facility (TFF) includes a fallback driver (SYS$FBDRIVER.EXE), a shareable image (TFFSHR.EXE), a terminal fallback utility (TFU.EXE), and a fallback table library (TFF$MASTER.DAT).

Note

TFFSHR has been removed from IMAGELIB because it is not a documented, user-callable interface. The image is still available in the SYS$LIBRARY: directory.

- To start TFF, invoke the TFF startup command procedure located in SYS$MANAGER, as follows:

  ```
  $ @SYS$MANAGER:TFF$SYSTARTUP.COM
  ```

- To enable fallback or to change fallback characteristics, invoke the Terminal Fallback utility (TFU), as follows:

  ```
  $ RUN SYS$SYSTEM:TFU
  TFU>
  ```

- To enable default fallback to the terminal, issue the following DCL command:

```
$ SET TERMINAL/FALLBACK
```

OpenVMS Alpha TFF differs from OpenVMS VAX TFF in the following ways:

- On Alpha systems, the TFF fallback driver is named SYS$FBDRIVER.EXE. On VAX systems, the TFF fallback driver is named FBDRIVER.EXE.

V6.1

- On Alpha systems, TFF is capable of handling 16-bit character fallback. The OpenVMS Alpha fallback table library (TFF$MASTER.DAT) contains four more 16-bit character tables than the VAX library. Table 3–1 describes these additional tables.

Table 3–1 TFF Character Fallback Tables

Table Name	Base	Description
BIG5_HANYU	BIG5	BIG5 for CNS 11643 (SICGCC) terminal/printer
HANYU_BIG5	CNS	CNS 11643 (SICGCC) for BIG5 terminal/printer
HANYU_TELEX	CNS	CNS 11643 for MITAC TELEX-CODE terminal
HANGUL_DS	KS	KS for DOOSAN 200 terminal

These tables are used mainly by the Asian region. Also, the table format was changed due to the support of 16-bit character fallback.

V1.0

- On Alpha systems, the TFU command SHOW STATISTICS does not display the size of the fallback driver (SYS$FBDRIVER.EXE).

RT terminals are not supported by TFF.

For more information about the Terminal Fallback facility, refer to the *OpenVMS Terminal Fallback Utility Manual*. ♦

3.33 VAX 7000 Systems (VAX Only)

VAX

This section contains release notes pertaining to VAX 7000 systems.

3.33.1 Problems and Restrictions

This section describes problems and restrictions experienced on VAX 7000 systems.

3.33.1.1 Dump Off System Disk (DOSD) Not Supported

V6.2

Dump Off System Disk (DOSD) functionality is not supported on VAX 7000 systems. Contact your Digital support representative for a remedial kit that will enable use of Dump Off System Disk on VAX 7000 systems. ♦

3.34 VMScluster Systems

The following sections contain information pertaining to VMScluster systems.

3.34.1 Changes and Enhancements

This section contains notes about changes to VMScluster systems.

3.34.1.1 SCSI VMScluster Systems

V6.2

Starting with OpenVMS Version 6.2, VMScluster systems support the Small Computer Systems Interface (SCSI) as a storage interconnect. See the *OpenVMS Version 6.2 New Features Manual* for a description of this feature, including configuration information, installation information, and a description of any known restrictions and problems.

3.34.1.2 Concurrency Improvements

V6.1

OpenVMS Version 6.1 and later provides an improved file system. This improved file system is also available as a special release known as the XQP+ for PATHWORKS.

One of the features provided by the improved file system is improved concurrency, which allows multiple processes to create files on a single disk in parallel.

In a VMScluster system, do not enable improved concurrency unless all nodes are running the improved file system. Each node in the cluster must be running one of the following:

- OpenVMS Alpha Version 6.1 or later
- OpenVMS VAX Version 6.1 or later
- XQP+ for PATHWORKS

If you are running the improved file system on all the nodes in your VMScluster, turn on improved concurrency by setting the static XQPCTL2 system parameter to 1.

3.34.1.3 FDDI Clusters (Alpha Only)

V6.1

Alpha

Beginning with OpenVMS Alpha Version 6.1, you no longer have to set system parameter PE3 to 1 to enable FDDI clustering. In fact, system parameter PE3 should not be set to any value other than 0. ♦

3.34.2 Problems and Restrictions

This section describes problems and restrictions pertaining to VMScluster systems.

3.34.2.1 KFESA/KFESB Controller Restrictions

V6.2

There are restrictions in the use of the KFESA/KFESB EISA DSSI controllers in systems that are members of a VMScluster system. This configuration will be supported only after a specific update is applied to your systems. This update may be present on your OpenVMS V6.2 CD–ROM distribution kit in the following location:

[KFESX]KFESX_UPDATE.BCK

To apply this update, restore the save set to a scratch directory and follow the instructions in the READ-ME.1ST file.

If this update save set is not present on your CD–ROM distribution kit, please contact your Digital support representative and ask for the TIMA kit associated with the KFESA/KFESB issue.

3.34.2.2 Booting Two Computers That Share a SCSI Bus (Alpha Only)

V6.2

Alpha

When you simultaneously boot two computers that share a SCSI bus, one of them might bugcheck early in the boot process with the following message:

```
** Code=00000215: MACHINECHK, Machine check while in kernel mode
```

However, the system that bugchecks will reboot immediately. Since the two systems are no longer booting simultaneously, the boot will succeed and the cluster will continue with no ill effects.

This problem is most common with AlphaServer 2100 systems that have more than one shared SCSI bus, but it can occur on other shared SCSI bus configurations. Digital expects to fix this software problem in a future release.

To avoid this problem, wait a few seconds between issuing boot commands to two systems sharing a SCSI bus. ♦

3.34.2.3 System Startup in a VMScluster Environment (Alpha Only)

V6.2

Alpha

In a VMScluster environment, under some circumstances the system startup procedure may fail to write a new copy of the ALPHAVMSSYS.PAR file. If this occurs, the console output from the boot sequence reports the following messages:

```
%SYSGEN-E-CREPARFIL, unable to create parameter file
-RMS-E-FLK, file currently locked by another user
```

This error creates an operational problem only when changing SYSGEN parameters using a conversational boot. For a normal, nonconversational boot, this error message is purely cosmetic because the parameter file has not changed. If a conversational boot is used, and SYSGEN parameters are changed at boot time, these changed parameters will be correctly used for the current boot of the system. However, since the boot procedure does not successfully write a new copy of the parameter file, these changed parameters will not be used in subsequent boots.

To permanently change SYSGEN parameters that have been changed by a conversational boot, run SYSGEN after the system has completed booting, and execute the following commands:

```
SYSGEN> USE ACTIVE
SYSGEN> WRITE CURRENT ♦
```

3.34.3 Corrections

This section describes corrections that affect VMScluster systems.

3.34.3.1 REMOVE_NODE Option in SYS$SYSTEM:SHUTDOWN.COM

V6.2

The REMOVE_NODE option in SYS$SYSTEM:SHUTDOWN.COM has been corrected to reduce cluster expected votes to a value that maintains quorum after the node shuts down. Quorum is maintained as long as at least one voting member remains in the cluster.

With this change to REMOVE_NODE you can now shut down any number of nodes in an OpenVMS cluster without encountering quorum loss. The benefit of this change is greatest for two-node clusters.

Digital strongly recommends that all nodes in a VMScluster environment have the SYSGEN parameter EXPECTED_VOTES set to the largest number of votes attainable in the cluster configuration.

This feature is provided only by the REMOVE_NODE option in SHUTDOWN.COM. Nodes that leave a cluster for other reasons, such as power failure, can still cause quorum loss.

To learn more about the impact of removing a node, refer to the *OpenVMS System Manager's Manual: Essentials* and the *VMScluster Systems for OpenVMS* manual.

3.35 VMSKITBLD.COM Procedure

This section contains notes pertaining to the VMSKITBLD.COM procedure.

3.35.1 Changes and Enhancements

This section describes a change in support for the VMSKITBLD.COM procedure.

3.35.1.1 VMSKITBLD.COM Not Supported on OpenVMS Alpha Version 6.2 (Alpha Only)

V6.2

Alpha

Using the VMSKITBLD.COM procedure to duplicate system files from an existing system disk to another disk now applies only to OpenVMS VAX systems (see the *OpenVMS System Manager's Manual: Essentials* for details).

Beginning with OpenVMS Alpha Version 6.2 systems, you can use the SYS$SYSTEM:AXPVMS$PCSI_INSTALL.COM procedure to install or upgrade from a running OpenVMS Alpha system to another target system disk. The *OpenVMS Version 6.2 New Features Manual* describes this new feature in more detail. ♦

3.35.2 Documentation Changes and Corrections

This section describes a documentation correction regarding the VMSKITBLD.COM procedure.

3.35.2.1 *OpenVMS System Manager's Manual: Essentials* (Alpha Only)

V6.2

Alpha

Section 2.5 of *OpenVMS System Manager's Manual: Essentials* incorrectly indicates that VMSKITBLD.COM is available on both OpenVMS VAX and OpenVMS Alpha systems. In fact, OpenVMS Alpha Version 6.2 no longer supports VMSKITBLD.COM. ♦

3.36 Volume Shadowing

The following sections pertain to volume shadowing software.

3.36.1 Changes and Enhancements

This section describes changes to volume shadowing software.

3.36.1.1 Mixed-Architecture VMScluster Changes

V6.2

Volume Shadowing for OpenVMS contains the following changes:

- Enabling of minimerge functionality in a mixed-architecture VMScluster environment that includes OpenVMS Alpha Version 6.1 nodes
- Support for geometry-based shadowing device recognition that allows devices, such as the RZ28 and RZ28B, to work in the same shadow set
- Software improvements that provide better interoperability for a mixed-architecture VMScluster environment

These changes are also available in a remedial kit that is distributed through Digital Customer Services worldwide. This kit applies to OpenVMS VAX Versions 5.5-2 and later and OpenVMS Alpha Version 6.1. For more information, please contact your Digital support representative.

3.36.1.2 New Bugcheck: SHADZEROMBR

V6.2

A synchronization problem which resulted in a virtual unit having zero members formerly caused ACCVIO system failures in Volume Shadowing for OpenVMS. Because having zero members is an illegal state for a mounted virtual unit, Volume Shadowing crashed the node in an attempt to maintain data integrity.

Version 6.2 of OpenVMS provides a new shadowing bugcheck called SHADZEROMBR. If this bugcheck occurs, please document and report what events occurred on other nodes in the VMScluster environment that also have the shadow set mounted. For example, if Node A and Node B share the same shadow set, report any problems that occurred on Node A before Node B failed with a SHADZEROMBR error. This will help OpenVMS Engineering determine what caused the failure.

To continue operation, reboot the failed node and remount the shadow sets.

3.36.1.3 Phase I Retirement (VAX Only)

V6.2

VAX

Starting with OpenVMS Version 6.2, the Volume Shadowing for OpenVMS product supports only phase II (host-based) shadowing. Phase I (controller-based) shadowing is no longer available. Phase II provides significantly enhanced data availability and supports a much wider range of configurations, disk controllers, and disks.

Any VAX nodes that still use phase I must be migrated to phase II in order to access shadowed data from an Alpha node on the cluster. For information about migrating from phase I shadowing to phase II, refer to the *Volume Shadowing for OpenVMS* manual. ♦

3.36.1.4 Timer-Based Polling

V6.2

Shadowing now uses a timer to adhere more accurately to the number of seconds specified by the SHADOW_MBR_TMO parameter. For directly connected SCSI devices that have been powered down or do not answer to polling, the elapsed time before a device is removed from a shadow set can still approach 1 minute. In all other situations, the elapsed time will closely approximate the number of seconds specified by the SHADOW_MBR_TMO parameter.

3.36.1.5 SCSI Shadowing Support (VAX Only)

V6.2

VAX

Volume Shadowing for OpenVMS Version 6.2 no longer supports shadowing operations for locally connected SCSI disks on the VAX 3000 and VAX 4000 series systems. ♦

3.36.2 Problems and Restrictions

The following sections describe known problems and other considerations pertaining to volume shadowing.

3.36.2.1 KDM70 Devices

V6.1 Volume Shadowing for OpenVMS (phase II) requires that KDM70 disk controllers run a minimum of Version 3.0 microcode.

3.36.2.2 DISMOUNT/CLUSTER Command Can Cause Clusterwide Hang

V6.1 When you issue the DISMOUNT/CLUSTER command to a shadow set, it can sometimes cause a clusterwide hang. If you experience this problem, use the DISMOUNT/SYSTEM command for every node in the VMScluster. You can enter this command once for every node in the VMScluster by using the SYSMAN DO command in combination with the DISMOUNT/SYSTEM command.

For more information about using the SYSMAN DO command, see the *OpenVMS System Manager's Manual*.

3.36.2.3 RZ57 Support Restricted in Shadow Sets

V6.1 OpenVMS systems support the RZ57 disk drive only with device revision level D01 and microcode revision 6000 or higher.

Disks that meet these conditions can be shadowed.

3.36.3 Documentation Changes and Corrections

This section contains a correction to the volume shadowing documentation.

3.36.3.1 *Volume Shadowing for OpenVMS* Manual

V6.1 The first bullet in Section 3.1 of the *Volume Shadowing for OpenVMS* manual incorrectly indicates that only Alpha systems support per-disk volume shadowing licensing. Starting with Version 6.1, OpenVMS VAX systems also support per-disk licensing.

4
Programming Release Notes

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

For information about new programming features included in OpenVMS Version 6.2, see the *OpenVMS Version 6.2 New Features Manual*.

4.1 Batch and Print Queues

This section contains release notes pertaining to batch and print queues.

4.1.1 Problems and Restrictions

This section describes problems and restrictions pertaining to batch and print queues. For problems important to system managers, refer to Section 3.4.1.

4.1.1.1 Terminating Executing Batch Jobs

V6.2

Under the following conditions, the DELETE/ENTRY command might fail to stop an executing batch job:

- The batch job is a DCL command procedure.
- There is an ON ERROR CONTINUE command (or SET NOON command) within the command procedure.

The DELETE/ENTRY command causes the job to terminate in phases. A delete_process AST routine is given in user mode, supervisor mode, and then executive mode. Because there is a small delay between each mode, it is possible that, in a batch job, a user-mode image might terminate and the command procedure might continue to execute until the supervisor-mode delete_process AST routine is executed.

The return status of the SYNCHRONIZE command is assumed to contain the termination status of the target batch job. In addition, command procedures would normally execute a command such as $ON ERROR THEN CONTINUE or $SET NOON before issuing the SYNCHRONIZE command. If a DELETE /ENTRY command is issued to the job executing the SYNCHRONIZE command, the JBC$_JOBABORT is interpreted as being the termination status of the target batch job rather than a return status of the SYNCHRONIZE command. The command procedure then continues to execute for a short period with this incorrect assumption and performs an operation such as requeuing the target batch job or incorrectly reporting a failure of the target batch job.

This problem has been fixed for the SYNCHRONIZE command by detecting this situation and waiting in an exit handler for longer than the delay between the user and supervisor mode termination delay.

Any other images that would report the job completion status obtained by the SJC$_SYNCHRONIZE_JOB function code of the $SNDJBC system service as the return status of the program should implement logic similar to the following:

1. Declare an exit handler
2. In the exit handler, implement the following logic:

```
IF (exit status is JBC$_JOBABORT)
THEN
    Wait 10 seconds
ENDIF
```

4.1.2 Corrections

This section describes corrections to batch and print queue features used by programmers. For batch and print queue corrections related to system management, see Section 3.4.2. For corrections of interest to the general user, see Section 2.1.1.

4.1.2.1 Accessing Batch Log Files

V6.2

In previous versions of OpenVMS, when a batch job was submitted without the /NAME qualifier, the log file name was not maintained. The log file was created when the job executed, but the log file name specification could not be accessed with the SYS$GETQUI system service or the F$GETQUI lexical.

This problem has been corrected.

4.1.2.2 Displaying Open Batch Jobs

V6.2

Beginning with OpenVMS VAX Version 5.5, open jobs were displayed when $GETQUI(display_entry) was used in a wildcard search because the undefined jobs (JOB$K_UNDEFINED) were being considered for display.

This problem has been corrected. Open jobs are no longer displayed when $GETQUI(display_entry) is used in a wildcard search.

4.1.2.3 $GETQUI Error Message

V6.2

Previously, when the $GETQUI service was called from a nonbatch process with the DISPLAY_JOB function and THIS_JOB flag, a JOBQUEDIS error was returned even when the queue manager was running. This inappropriate error was also returned when a process was spawned from a batch process.

This problem has been fixed and the appropriate error (JBC-E-NOSUCHJOB) is now returned.

4.1.2.4 Log File Defaults for Batch Jobs

V6.2

In previous versions, if a JOBNAME included invalid characters in a file specification and any of the components of the log file name were drawn from defaults, the batch job failed with the following error message:

```
%RMS-F-SYN, file specification syntax error
```

This problem occurred when the $SNDJBC system service was used with both the SJC$_LOG_SPECIFICATION and the SJC$_JOB_NAME item codes. If the log file name was not a complete file specification, defaults were applied incorrectly to create one.

This problem has been corrected, and the job no longer fails.

4.1.2.5 Restarting FORTRAN Carriage Control Files

V6.2 In previous versions, when a paused or stopped queue printing a FORTRAN carriage control file was restarted with the /TOP_OF_FILE qualifier, extraneous formatting characters were sent to the printer device. Usually, an extra line feed was output to the printer.

This problem has been fixed, and the /TOP_OF_FILE qualifier can be used when restarting queues printing FORTRAN carriage control files.

4.1.2.6 Symbiont BADLOGIC Dump After STOP/QUEUE/RESET and START/QUEUE

V6.2 In previous versions of OpenVMS, a STOP/QUEUE/RESET command that was followed quickly by a START/QUEUE command caused the print symbiont to produce a BADLOGIC dump.

A workaround for this problem stated that you should wait at least 1 minute between entering STOP/QUEUE/RESET and START/QUEUE commands.

This problem has been corrected. You no longer need to wait between entering the two commands.

4.1.2.7 TASK_STATUS Requests to the Queue Manager

V6.2 In previous versions, when a symbiont sent a TASK_STATUS request to the queue manager and there were no active jobs, the request was rejected with the following message:

```
Message from user QUEUE_MANAGE on CRATER
%QMAN-I-INVSMBMSG, invalid data in message from symbiont on queue
TEST is being ignored
```

However, device status can change when no jobs are active (for example, SMBMSG$V_STALLED or SMBMSG$V_UNAVAILABLE conditions are not tied to a job). Therefore, this situation has been corrected. Now, when a symbiont sends a TASK_STATUS request to the queue manager and there are no active jobs, the request is accepted.

4.2 Debugger

This section contains release notes pertaining to the OpenVMS Debugger.

Unless specified otherwise, the release notes apply to both the character-cell and DECwindows Motif interfaces of the debugger.

4.2.1 Changes and Enhancements

This section describes changes and enhancements in OpenVMS Debugger Version 6.2.

4.2.1.1 Performance Improvements

V6.2 Performance improvements have been made in the following areas:

- Previously, in the DECwindows Motif interface, lengthy examines resulted in performance degradation. Performance has now been considerably improved through optimized text output routines. In addition, a new message view pop-up menu command named CLEAR COMMAND WINDOW (which entirely clears the command window of its current contents) has been added to restore performance after lengthy output.
- The previous performance degradation in the debugger startup and in the SET MODULE, SET SCOPE, DECLARE, and SHOW SYMBOL commands has been corrected.

Alpha

- On OpenVMS Alpha systems, previously the STEP/RETURN command was very slow. Performance has now been considerably enhanced. ♦

4.2.1.2 VMSDEBUG.DAT File

V6.2

This note pertains only to the DECwindows Motif interface.

When you have installed the OpenVMS Debugger, be aware that the VMSDEBUG.DAT file has been revised, and check for the existence of an old file named VMSDEBUG.DAT in your directory. If you do not have a VMSDEBUG.DAT file in your login directory when you run the debugger, the debugger will start up with the full complement of definitions found in the system copy of this file, DECW$SYSTEM_DEFAULTS:VMSDEBUG.DAT. There is nothing more for you to do.

If you do have an old copy of VMSDEBUG.DAT in your login directory, the debugger will use whatever definitions are in it. Because some changes have occurred, there may be some minor visual differences, particularly in the definitions of fonts. If you have made customizations to your GUI and then saved the customizations from the menu function, the current design replaces the new VMSDEBUG.DAT with an older format in your SYS$LOGIN directory, and you will lose the additional features of the latest version. It is therefore recommended that you take advantage of the latest version of the VMSDEBUG.DAT file by doing one of the following:

– Delete your old SYS$LOGIN:VMSDEBUG.DAT, thus defaulting to the system copy.

– Copy the DECW$SYSTEM_DEFAULTS:VMSDEBUG.DAT file and carefully merge the changes with your SYS$LOGIN version.

4.2.2 Problems and Restrictions

This section describes known debugger problems and restrictions.

4.2.2.1 Behavior After Stack Corruption

V6.2

The debugger allocates a certain amount of memory at startup and shares the stack with the user's program. If a user process exception results in exhaustion of resources or corruption of the stack, the debugger may be incapable of regaining control, and the debug session may terminate.

You may be able to work around this problem by trying one of the following:

- Change your source code, temporarily or permanently, to reduce resource consumption or lessen the use of stack space
- Increase quotas
- Specify a larger stack size when linking your program

Be aware of this potential behavior after the occurrence of stack corruption messages or warnings about continuing from a severe error. In either case, the integrity of the debug session cannot be guaranteed.

4.2.2.2 CONNECT Command

V6.2

You cannot use the CONNECT command to connect to a subprocess of a process running under debugger control.

To work around this restriction, use the SET PROCESS command to connect to subprocesses.

4.2.2.3 Detached Processes

V6.2

You cannot use the DECwindows Motif interface to the debugger to debug detached processes, such as print symbionts, which run without a Command Line Interpreter (CLI).

The debugger's DECwindows Motif interface requires either the two-process or multiprocess debugger configurations so that the main debugger image can communicate with the debugger's UI image. These debuggers require you to start the program you are debugging from the CLI; the debuggers themselves are created through a call to LIB$SPAWN, which requires a CLI.

If you want to debug a detached process that runs without a CLI, use the character-cell (screen mode) interface to the single-process debugger.

4.2.2.4 SET TASK/ACTIVE Command

V6.2

The debugger command SET TASK/ACTIVE does not work for DECthreads.

Instead, you can use the following effective substitutes on DECthreads:

- For query-type actions, use the SET TASK/VISIBLE command.
- To gain control to step through a particular thread, use strategic placement of breakpoints.

4.2.2.5 SPAWN/NOWAIT Command

V6.2

If your application attempts to spawn a subprocess from a process running under debugger control and the following conditions apply, the debugger hangs:

- You started the spawned subprocess with a RUN/DEBUG command
- The Nowait flag is clear

This occurs because the wait direction causes the current debug process to hibernate until the spawned process completes. The spawned process never completes because the debug kernel image spawned with your spawned subprocess waits indefinitely for communication from the debug main image. The debug main image cannot communicate due to the wait direction.

To work around this restriction, do one of the following:

- Start the spawned subprocess with the RUN/NODEBUG command and then issue the SET PROCESS command.
- Set the Nowait flag.

4.2.2.6 System Service Interception and Shared Linkage Images (Alpha Only)

V6.2

Alpha

On OpenVMS Alpha systems, anything that uses system service interception (SSI), like the debugger or the heap analyzer (HA), is unable to intercept system service call images activated by shared linkage. The image activator, therefore, avoids shared linkage for images linked or run with /DEBUG, and instead activates private image copies.

This has performance implications for user applications under debugger or HA control, as images activated by shared linkage run faster. ♦

4.2.2.7 DEBUG.EXE Image (VAX Only)

V6.1

VAX

On OpenVMS VAX systems, you cannot debug an image named DEBUG.EXE. This name is used internally for the kernel portion of the debugger. To work around this problem, do the following:

```
$ DEFINE LIB$DEBUG SYS$SHARE:DEBUG
```

Optionally, you can rename your image to something other than DEBUG.EXE. ♦

4.2.2.8 STEP/OVER Steps into Fortran Run-Time Library Routines (VAX Only)

V6.1

VAX

On OpenVMS VAX systems, if you are debugging a DEC Fortran program, and the debugger encounters a Fortran I/O operation such as READ, WRITE, or PRINT, you will notice that the Fortran compiler calls a Fortran Run-Time Library routine to complete the I/O operation. When you issue a STEP command at the call to the RTL routine, the debugger steps into, rather than over, the routine and issues the following error:

```
%DEBUG-W-NOSCRLIN, no source line for address nnnnnnnn
```

The problem occurs because the debugger steps across routines by searching for a RET instruction to be executed, and, in this case, the Fortran compiler is deallocating temporary strings from the stack in a nonstandard way, without explicit RET instructions.

To recover from the error message, issue several STEP commands to return the session to the expected line of code. To prevent recurrence of the problem, modify your program to include a temporary variable that stores the results of the function prior to the I/O statement. For example:

```
    CHARACTER*23 c1, c2
    c1 = c2()                   ! c1 is the temporary variable
    WRITE (*) c1
    END
C
    CHARACTER*23 FUNCTION c2()
    c2 = 'ABCDEFGHIJKLMNOPQRSTUVW'
    RETURN
    END ♦
```

4.2.2.9 Watchpoint Support

VAX V6.0
Alpha V1.0

The following are problems and restrictions with the debugger's support for watchpoints:

- Watchpoints set on variables whose addresses are in global sections do not work. Attempting to set a watchpoint on a location in a global section will result in a %DEBUG-E-BADWATCH message.
- If a watched location changes during a system service routine, you will be notified, as usual, that the watchpoint occurred. However the stack will show one or more debugger frames on top of the frame or frames for your program.

  To work around this problem, enter one or more STEP/RETURN commands to get back to your program.

4.2.2.10 CALL and SHOW CALLS Commands (Alpha Only)

V1.5

Alpha

On OpenVMS Alpha systems, the following CALL command problems and restrictions appear in this release:

- You cannot pass floating-point parameters by value in a CALL command.
- After you issue a SHOW CALLS command, the output may include system call frames in addition to the user call frames associated with your program. System call frames appear in the following circumstances:
  - When an exception occurs
  - When an asynchronous system trap occurs
  - When a watchpoint occurs in system space

  The display of system call frames does not indicate a problem. ♦

4.2.2.11 DECwindows Motif Interface

V1.5

The following problems or restrictions are specific to the DECwindows Motif interface:

- Occasionally, if you are debugging a UI application and you have many debugger windows overlapping the user program's windows, the X server will abruptly terminate the user program.

  To avoid this problem, refrain from overlapping or covering windows belonging to the user program.
- If you are stopped at a breakpoint in a routine that has control of the mouse pointer by a PointerGrab or a KeyboardGrab, you will hang your workstation.

  To work around this problem, debug your program using two workstations. For more information, see the *OpenVMS Debugger Manual*.
- Occasionally, if you are using DECwindows Motif Version 1.1 and are scrolling or clicking on items in the Register View, the following message is displayed in the DECterm from which you initiated the debugging session:

  ```
  X Toolkit Warning: Not all children have same parent
     in XUnmanageChildren
  ```

  This error can be safely ignored.
- Table 4–1 lists debugger commands that are disabled in the DECwindows Motif interface. The debugger issues an error message if you try to enter any of these disabled commands at the command prompt, or when the debugger executes a command procedure containing any of these commands.

Table 4–1 Debugger Commands Disabled in the DECwindows Motif Interface

ATTACH	SELECT
CANCEL MODE	(SET, SHOW) ABORT_KEY
CANCEL WINDOW	(SET, SHOW) KEY
DEFINE/KEY	(SET, SHOW) MARGINS
DELETE/KEY	SET MODE [NO]KEYPAD

(continued on next page)

Table 4–1 (Cont.) Debugger Commands Disabled in the DECwindows Motif Interface

DISPLAY	SET MODE [NO]SCREEN
EXAMINE/SOURCE	SET MODE [NO]SCROLL
EXPAND	SET OUTPUT [NO]TERMINAL
EXTRACT	(SET, SHOW) TERMINAL
HELP	(SET, SHOW) WINDOW
MOVE	(SHOW, CANCEL) DISPLAY
SAVE	SHOW SELECT
SCROLL	

- DECwindows Motif does not provide for specialized key support (such as Ctrl/Y), but the DECwindows Motif interface provides alternative means of executing these functions (for example, the STOP button).
- Commands related to character-cell terminal display apply only to the command interface. These commands are disabled in the DECwindows Motif interface.
- If you have not installed a DECwindows Motif license (DW-MOTIF) on the node from which you are using the debugger, the debugger's GUI fails with the following error messages:

  ```
  %NONAME-W-NOMSG, Message number 00000000
  -DEBUG-F-FATALSTATUS, a fatal condition was detected by the debugger.
  %DEBUG-F-INITERR, an error has occurred during debugger initialization,
  unable to continue this session.
  ```

  To return to DCL level after this failure, enter a Ctrl/C or Ctrl/Y key sequence. To correct the problem, install a DECwindows Motif license or use the character-cell interface by defining DBG$DECW$DISPLAY to " ".
- The DECwindows Motif interface to the debugger does not use a DECterm. Therefore, commands that require a DECterm are disabled in the DECwindows Motif interface. The exception is the EDIT command, which is still available.
- Occasionally, the DECwindows Motif interface does not accept properly formatted Fortran commands. For example, if you select the fragment "qdata(:ilen)" from the program statement "read(qdata(:ilen),100)ivalue" with the mouse and paste it into an EXAMINE command at the DBG> prompt, the debugger issues an error message:

  ```
  DBG>  EXAMINE qdata( :ilen)
  DEBUG-E-MISOPEMIS, misplaced operator or missing operand at 'end of
  expression'
  ```

  To work around this problem, reenter your command:

  ```
  DBG>  EXAMINE qdata(1:ilen)
  ```

4.2.2.12 EVALUATE Command (Alpha Only)

V1.5

Alpha

On OpenVMS Alpha systems, when you issue the EVALUATE command for an integer, the debugger truncates the return value if it is larger than a longword. ♦

4.2.2.13 EXAMINE LABEL[*n*] Command (Alpha Only)

V1.5

Alpha

On OpenVMS Alpha systems, a command in the form EXAMINE LABEL[*n*] or EXAMINE LABEL(*n*), where LABEL is a label for a code location and *n* is an integer, causes an access violation error. In this case, the debugger does not handle the error.

Note that this problem does not occur when the label marks the start of data storage, as in a MACRO program. ♦

4.2.2.14 Kept Debugger

V1.5

The following problems or restrictions are specific to the Kept Debugger:

- If a previous debugger process has not completely stopped, you may see the following error at debugger startup:

  ```
  %DEBUG-E-INTERR, internal debugger error in
     DBGMRPC\DBG$WAIT_FOR_EVENT got an ACK
  ```

  To fix this problem, exit the debugger. Then use the DCL command SHOW PROCESS/SUBPROCESS to check if any debugger subprocesses exist. If so, you can stop them by using the DCL command STOP. You should then be able to restart the debugger without seeing the error described above.

- Ctrl/Y–DEBUG is not supported in the Kept Debugger configuration.

- Running a sequence of many large programs may cause the debugger to fail because it has run out of memory, global sections, or some other resource.

  To fix this problem, exit the debugger and restart the debugging session.

- Many commands are disabled when there is no running program. This includes commands that might be expected to work, such as SET STEP and SET PROMPT/SUFFIX. The disabled command may cause DBG$INIT files to generate %DEBUG-W-NOPROGRAM messages. These commands are enabled once a RUN command has been executed.

- The prompt may change when a RUN command is executed. It will change back to its former state once the program has completed.

- If you are using the DECwindows Motif interface (as opposed to character-cell screen mode), and you try to run a program that does not exist, or misspell the name of a program that does exist, you may not notice the following error messages displaying:

  ```
  %DCL-W-ACTIMAGE, error activating image
  -CLI-E-IMAGEFNF, image file not found
  ```

  The reason is that DECwindows Motif displays the messages in the DECterm window, rather than in the Command View. Therefore, it is not always obvious that an error has occurred.

  To avoid this problem, make sure the "Select an application to run" box in the File Selection popup contains a valid file specification.

- The %DEBUG-I-INITIAL is not displayed after execution of the RERUN /SAVE command. The absence of this message does not adversely affect the execution of this command.
- The Kept Debugger shares I/O channels with the parent process when it is run by a SPAWN/NOWAIT command. Therefore, you must press the Return key twice on the DECterm from which the debugger was run after the debugger version number has appeared in the Communications Pane.

  Optionally, you can execute the Kept Debugger in the following manner:

```
$ DEFINE DBG$INPUT NL:
$ SPAWN/NOWAIT RUN DEBUG/KEEP
```

- If you issue the RERUN command while your file (that is, the image you wish to rerun) is locked by another user, the debugger returns the following message:

```
%DEBUG-E-NORERUNPGM, There is no program to RERUN
```

  This situation might occur, for example, if another user has changed the protection on the file. To work around this problem, have your system manager change the protection. Then, issue a RUN command and reset breakpoints.

4.2.2.15 SET BREAK/UNALIGNED_DATA Command and Related System Service Call (Alpha Only)

V1.5

Alpha

On OpenVMS Alpha systems, the SET BREAK/UNALIGNED_DATA command calls the SYS$START_ALIGN_FAULT_REPORT system service routine. Do not issue this command if the program you are debugging includes a call to the same SYS$START_ALIGN_FAULT_REPORT routine. If you issue the command before the program call, the program call fails. If the program call occurs before you issue the command, unaligned breaks are not set. ♦

4.2.2.16 Complex Variables in Fortran Programs (Alpha Only)

V1.0

Alpha

On OpenVMS Alpha systems, the debugger cannot evaluate expressions that contain complex variables. (Currently, DEC Fortran is the only supported language that provides complex variables.) To work around this problem, examine the complex variable and then evaluate the expression using the real and imaginary parts of the expression obtained from the EXAMINE command. ♦

4.2.2.17 Concealed Rooted-Directory Logical Names

V1.0

If you use a rooted-directory logical name to specify the location of a source file when compiling a program with the /DEBUG qualifier, make sure that the rooted-directory logical name is concealed. You must include the /TRANSLATION_ATTRIB=CONCEALED qualifier in your logical name definition, as follows:

```
DEFINE /TRANSLATION_ATTRIB=CONCEALED root_dir_log_name disk:[dir.]
```

If the rooted-directory logical name is not concealed, and you move the source file to another directory after compilation, you cannot use the debugger SET SOURCE command to specify the new location of the source file.

4.2.2.18 DEPOSIT/TYPE Command with C Programs

V1.0

When debugging a C program, you cannot use the DEPOSIT/TYPE command if the type specified is a mixed or lowercase name. For example, suppose the program has a function like the following:

```
xyzzy_type foo ()
{
 xyzzy_type     z;
 z = get_z ();
 return (z);
 }
```

If you try to enter the following command, the debugger issues a message that it cannot find the type "xyzzy_type":

```
DBG> DEPOSIT/TYPE=(xyzzy_type) z="whatever"
```

4.2.2.19 $HIBER Call

V1.0

In the following cases, a user application remains in HIB (hibernate) state:

- If a program, running under the two-process or multiprocess debugger, issues a $WAKE call followed by a $HIBER call, the user application hibernates.
- If you step inside or interrupt RTL or system services routines that make use of $WAKE or $HIBER, for example Lib$wait or sys$schdwk, the user application hibernates.
- When an application includes the LCK$M_DEQALL modifier in a $DEQ system service call, this modifier breaks communication links between the portion of the debugger in the user process (the kernel) and the debugger main process. The result is that the user's process stays in hibernate (HIB) state.

  To work around this problem, debug these application programs using the limited one-process mode, rather than the default or the multiprocess mode. To set up one-process mode, issue the following command:

  ```
  $ DEFINE DBG$PROCESS NONE
  ```

4.2.2.20 Inlined Routines (Alpha Only)

V1.0

Alpha

On OpenVMS Alpha systems, the debugger does not support attempts to debug inlined routines. If you attempt to debug an inlined routine, the debugger issues a message that it cannot access the routine, as shown in the following example:

```
%DEBUG-E-ACCESSR, no read access to address 00000000
```

To work around this problem, compile your program with the /NOOPTIMIZE qualifier. ♦

4.2.2.21 Null Frame Procedures and SHOW CALLS (Alpha Only)

V1.0

Alpha

On OpenVMS Alpha systems, there is a problem with null frame procedures that prevents the SHOW CALLS display from including their frames. For more information on null frame procedures, see the *OpenVMS Calling Standard*. ♦

4.2.2.22 SHOW BREAK and SHOW TRACE Commands (Alpha Only)

V1.0

Alpha

On OpenVMS Alpha systems, SHOW BREAK and SHOW TRACE commands do not display individual instructions when the break or trace is on a particular class of instruction (such as SET BREAK/CALL or SET BREAK/RETURN). ♦

4.2.2.23 STEP/INTO Command and User Exception Handlers (Alpha Only)

V1.0

Alpha

On OpenVMS Alpha systems, when execution is stopped at an exception break, STEP/INTO does not transfer control to a user exception handler. To work around this problem, set a breakpoint on the handler. ♦

4.2.2.24 STEP/OVER Command Error with One-Line Program Loops (Alpha Only)

V1.0

Alpha

On OpenVMS Alpha systems, when you issue the STEP/OVER command at a program loop that is coded on a single source line, and that source line also contains a routine call, the debugger steps into the called routine instead of stepping to the next source line. In the following example, if you issue the STEP /OVER command when execution is stopped at the FOR loop, the debugger steps into the square routine instead of stepping to the j assignment statement:

```
for (i=0;i<10;i++) square(i);
j=6;
```

To work around this problem, either set a temporary breakpoint on the line following the FOR loop (in the previous example, j=6), or move the routine call to a separate line, as follows:

```
for (i=0;i<10;i++)
   square(i); ♦
```

4.2.2.25 Translated Images (Alpha Only)

V1.0

Alpha

On OpenVMS Alpha systems, the debugger does not support attempts to debug translated images. If you must debug a translated image, use the Delta/XDelta Debugger. For more information on this debugger, see the *OpenVMS Delta /XDelta Debugger Manual.* ♦

4.2.2.26 Vector Support (VAX Only)

V5.4

VAX

On OpenVMS VAX systems, the following are problems and restrictions with the debugger's support for vectorized programs:

- When the programming language is BLISS, COBOL, or RPG, you must specify a type qualifier to deposit into %VMR. For example:

  ```
  DEPOSIT/QUADWORD %VMR = %HEX 0FFFFFFFF
  ```

- When the programming language is PL/I, COBOL, or DIBOL, the command EXAMINE %VMR displays %VMR as an array of bits instead of as a hexadecimal quadword. Enter the command EXAMINE/HEX/QUADWORD %VMR to obtain the default behavior for other programming languages.

- When the vector mode is synchronized (that is, if you have entered the command SET VECTOR_MODE SYNCHRONIZED), the debugger suspends execution twice at any breakpoints that were set on vector instructions. To resume execution from such breakpoints, you must enter the GO or STEP command twice. ♦

4.2.3 Corrections

This section describes corrections to problems that existed in OpenVMS Debugger Version 6.1.

4.2.3.1 Breakpoints (Alpha Only)

V6.2

Alpha

On OpenVMS Alpha systems, a problem that caused missed breakpoints has been corrected. ♦

4.2.3.2 Breakpoints and Display of Source and Scope

V6.2

Previously, in the DECwindows Motif interface, the display of the source and scope after a breakpoint event was sometimes incorrect. This problem has been corrected, and the scope and source are now kept current and correctly displayed.

4.2.3.3 COBOL: STEP and GO

V6.2

Previously, in COBOL programs, a STEP and GO sometimes resulted in %DEBUG-E-ERROR_BLOCK and %DEBUG-I-SS_INT_END. This problem has been corrected.

4.2.3.4 Communications Pane Sizing

V6.2

Previously, in the DECwindows Motif interface, if you decreased the size of the Communications Pane so that the prompt was occluded, the message area was not automatically scrolled to display the prompt, and you had to use the vertical scroll bar to scroll the view so that the prompt reappeared. This problem has been corrected.

4.2.3.5 DEBUG.EXE Image (Alpha Only)

V6.2

Alpha

Previously, on OpenVMS Alpha as well as OpenVMS VAX systems, you could not debug an image named DEBUG.EXE (a name used internally for the kernel portion of the debugger). This problem has been corrected on OpenVMS Alpha systems. ♦

4.2.3.6 Edit Save Function

V6.2

Previously, in the DECwindows Motif interface, the Edit Save function sometimes wrote a corrupt output file. This problem has been corrected.

4.2.3.7 EVALUATE Command

V6.2

In previous releases, the debugger would generate an OPTNOTALLOW error when the Pascal IN operator was used on a packed array in an EVALUATE command. This problem has been corrected.

4.2.3.8 EXAMINE Command on SCAN Tree Variables (VAX Only)

V6.2

VAX

In previous versions of OpenVMS VAX, an ACCVIO status resulted from attempts to use the EXAMINE command on SCAN tree variables. This problem has been corrected. ♦

4.2.3.9 Hebrew Command View Messages

V6.2 Previously, for the DECwindows Motif interface (V1.2), in the Hebrew version, command view messages were compressed into a single line. This problem has been corrected.

4.2.3.10 Include Files in a Text Library

V6.2 Previously, when the debugger accessed an include file kept inside a text library, it prevented user applications from updating/replacing the include file by means of the text library. This problem has been corrected.

4.2.3.11 Monitor View Display of Long Strings

V6.2 Previously, in the DECwindows Motif interface, the Monitor View failed to display long strings. This problem has been corrected so that the debugger now truncates the display of long strings in the Monitor View at 256 characters, and reports this with an informational message.

4.2.3.12 Multidimensional Arrays

V6.2 In previous releases, sometimes the debugger incorrectly examined nonzero-based multidimensional arrays. This problem has been corrected.

4.2.3.13 PL/I Controlled Variables

V6.2 In previous releases, the debugger generated an internal error on attempts to examine PL/I controlled variables that were arrays. This problem has been corrected.

4.2.3.14 Process Dump Files

V6.2 Previously, the debugger could not analyze a process dump file if the dump described an exception where the program counter (PC) was zero. This problem has been corrected.

4.2.3.15 Radixes for Monitored Items

V6.2 Previously, if you attempted to typecast or change the radix for a monitored item (for example, a task in a multitasking program) where deposit operations did not make sense, the debugger did not issue an error message. Instead, it tried to complete the operation and then froze the display. This problem has been corrected.

4.2.3.16 Records with the Same Symbol Name

V6.2 Previously, when more than one record used the same symbol name, an examine sometimes provided information for the wrong record, even if fully scoped. This problem has been corrected.

4.2.3.17 RERUN/SAVE Command

V6.2 In previous releases, 'EVENT=' type breakpoints were not saved during execution of a RERUN/SAVE command. This problem has been corrected.

4.2.3.18 SET RADIX Command

V6.2 When you use the SET RADIX command, the radix you specify now correctly persists across runs, reruns, set language, set image, and so forth.

4.2.3.19 SET SOURCE Command (VAX Only)

V6.2

VAX

In Version 6.1 of the OpenVMS VAX debugger, the default behavior of the SET SOURCE command was inadvertently changed to SET SOURCE/LATEST. The default setting of the SET SOURCE command has been restored to SET SOURCE /EXACT, as it was in OpenVMS VAX Version 6.0 and previous versions of the debugger. ♦

4.2.3.20 User-Written System Service Calls (Alpha Only)

V6.2

Alpha

On OpenVMS Alpha systems, a problem involving stepping over CHM* and "misdirected" user-written system service calls has been corrected. ♦

4.2.3.21 Watchpoint Reporting

V6.2

In previous releases, the debugger did not always report watchpoints. This problem has been corrected.

4.2.3.22 Watchpoint Setting on Bit Fields

V6.2

In previous releases, in the DECwindows Motif interface, the debugger sometimes failed to set watchpoints on a bit field. This problem has been corrected.

4.3 DEC C++

The following sections contain release notes pertaining to DEC C++.

4.3.1 Problems and Restrictions

This section describes known DEC C++ problems and restrictions.

4.3.1.1 SYS$LIB_C.TLB Structures Need Prototypes (Alpha Only)

V6.1

Alpha

DEC C++ users who include header files from SYS$LIB_C.TLB may receive compiler errors. Many of the structures within SYS$LIB_C.TLB contain pointers to other structures. These pointers are defined using the tag name of the structure to which it points. In C, if you declare something using a tag name, it is not necessary to predefine it. DEC C++ requires that the name be predefined. If the header file you are including contains any pointers to structures, it will be necessary to create structure prototypes for them.

As an example, the inclusion of a header file (header.h) that contained the following:

```
typedef struct  _structa {
    struct _structb    *ptr1;
    struct _structc    *ptr2;
} STRUCTA;
```

would require the following structure prototypes defined before the #include for the header file:

```
struct  _structb;
struct  _structc;

#include <header.h>
```

♦

4.4 DEC C++ Class Library

This section contains release notes pertaining to the DEC C++ class library.

4.4.1 Changes and Enhancements

This section describes enhancements to the DEC C++ class library.

4.4.1.1 Object Library Format

V6.2 The DEC C++ class library has always been provided in shareable image format. Starting with OpenVMS Version 6.2, the DEC C++ class library is also provided in object library format. Please refer to the *OpenVMS Version 6.2 New Features Manual* for details about linking using this object library.

4.4.2 Corrections

V6.2 The following problems that existed in the DEC C++ class library in OpenVMS Version 6.1 have been corrected:

- An access violation is no longer generated when your application writes into an uninitialized ostream_withassign. Instead, the written text is simply ignored.

VAX

- The class library image supplied with OpenVMS VAX Version 6.1 and with the ECO for DEC C/C++ Run-Time Components for VAX/VMS (AACRTVAXE02060 or CSCPAT_1150012) incorrectly defined the size of a strstreambuf to be several bytes larger than the defined header file. So if your application declared a strstreambuf, the strstreambuf constructor and several other strstreambuf functions would write into the memory beyond that allocated for your strstreambuf. The consequences were unpredictable.♦

- An access violation no longer occurs when a null pointer is passed to the insertion operator (<<) within the ostream class.

VAX

- The dynamic expansion of strstream and ostrstream buffers has been corrected on OpenVMS VAX. Previously, when data was inserted that overflowed the buffer, the expansion of the buffer could fail even if memory was available. This resulted in the buffer contents being truncated. ♦

- A file-positioning problem for bidirectional fstreams has been corrected. Previously, if your application switched from reading an fstream to writing an fstream, with or without an intervening seek operation, the necessary synchronization of the external file position with the get pointer was often skipped. This resulted in the written data being placed incorrectly within the file.

- The real( ) function within the stopwatch class was incorrectly returning the CPU time. Now it returns the clock time as documented.

- The >> operator within the istrstream class now works correctly when the conversion base format is hex and hexadecimal values are read that begin with "0" but are not prefixed with "0x."

- The flush( ) function within the ostream class was incorrectly calling overflow( ). Now flush( ) calls sync( ) as documented.

- The ignore( ) function within the istream class now works correctly. Previously it would ignore two and only two characters.

- The input functions within the stdiostream class no longer skip every other character.

- The sync_with_stdio( ) function within the ios class no longer causes subsequent istream operations to skip every other character.
- The SYS$OUTPUT and SYS$ERROR files are no longer opened during the initialization of cout and cerr, respectively. Previously this behavior resulted in an empty file being created when SYS$OUTPUT or SYS$ERROR was defined to be a file and the application generated no output.
- The showpoint flag within the ios class now forces the decimal point to be displayed.

4.5 DEC C Run–Time Library

The following sections contain release notes pertaining to the DEC C Run-Time Library (RTL).

4.5.1 Changes and Enhancements

This section describes changes and enhancements to DEC C RTL software.

4.5.1.1 Floating-Point Values Now Conform to IEEE Standard

V6.2

The `ecvt`, `fcvt`, and `gcvt` functions were modified to represent the following values specified in the IEEE Standard for floating-point arithmetic:

Value	Representation
Quiet NaN	NaNQ
Signaling NaN	NaNS
+Infinity	Infinity
-Infinity	-Infinity

The sign associated with each of these values is stored in the sign parameter of `ecvt`, `fcvt`, and `gcvt`. Also note that in IEEE Floating Point, a value of zero can be positive or negative, as set by the sign parameter.

4.5.1.2 Getenv Function Now Performs Case-Insensitive Lookup

V6.2

The function `getenv` was enhanced to perform a case-insensitive lookup if the case-sensitive lookup fails. In most instances, logical names are defined in uppercase. The `getenv` function now can find logicals that include lowercase letters in the name.

4.5.1.3 Internationalization Support

V6.2

The DEC C RTL has added capabilities to allow application developers to create international software. The DEC C RTL obtains information about a language and a culture by reading this information from *locale* files.

If you are using these DEC C RTL capabilities, you must install a separate kit to provide these files to your system. The save set, VMSI18N062, is provided on the same media as OpenVMS Version 6.2.

To install this save set, follow the standard OpenVMS installation procedures using this save set name as the name of the kit. There are four categories of locales that you can select to install. You may select as many locales as you need by answering the following prompts:

```
Do you want European and US support?
Do you want Chinese support?
Do you want Japanese support?
Do you want Korean support?
```

This kit also has an Installation Verification Procedure that Digital recommends you run to verify the correct installation of the kit.

4.5.1.4 Math Functions Conform to XPG4 (VAX Only)

V6.2

VAX

The following changes have been made to DEC C RTL in OpenVMS VAX Version 6.2 so that the behavior now matches that on OpenVMS Alpha systems.

- The behavior of the log, log10, and pow functions now conforms to the behavior specified by XPG4. As such, log and log10 of zero return -HUGE_VAL and set errno to ERANGE; pow of zero with a negative exponent returns -HUGE_VAL and sets errno to EDOM; and pow of zero with a zero exponent returns 1.0. ♦

4.5.1.5 Open Routines Now Validate RMS Options

V6.2

The open, fopen, and creat functions no longer ignore invalid RMS options. If an unrecognized RMS option is passed as a parameter to these functions, a return value of -1 is returned and errno is set to EINVAL (Invalid argument).

4.5.1.6 Setvbuf Function

V6.2

The setvbuf function has been changed to give it more functionality and make it more compatible with the setvbuf function on other operating systems.

Prior to this release, if the buffer argument to setvbuf was NULL, setvbuf would reset the buffer to the one allocated by the DEC C RTL when the file was opened. Starting with Version 6.2, if a NULL buffer argument is passed and the size argument is larger than the buffer allocated by the RTL when the file was opened, setvbuf allocates a buffer equal to the specified size and uses that as the file buffer.

4.5.1.7 Sleep Function Now Returns Unslept Seconds

V6.2

The return value of the sleep function has been changed to be in line with the System V Interface Definition and the X/Open Portability Guide, Release 4. The return value is the number of seconds that it prematurely awoke, due to the delivery of a signal or a call from the sys$wake function. A return value of zero is returned when a sleep call waits the entire time period. Prior to these changes, the sleep function returned the number of seconds that were originally requested.

4.5.1.8 Wildcards No Longer Accepted by stat Function

V6.2

The stat function no longer accepts wildcard characters as part of the file specification. If a wildcard character is found, stat returns -1 and sets errno to ENOENT (No such file or directory).

4.5.1.9 Writing to Pipes Now Conforms to POSIX Standards

V6.2

The DEC C RTL now properly deals with EOF when reading and writing to pipes. The correct behavior is described in the IEEE standard P1003.1-1988, Section 6.4.1.2, which states that an EOF is written to a pipe only when no writers are left. Previously, the DEC C RTL wrote an EOF to a pipe every time a writer closed a pipe.

The DEC C RTL no longer incorrectly writes an EOF to a pipe when the user issues a write of zero bytes. The corrected behavior now conforms to the IEEE Standard P1003.1-1988, Section 6.4.1.2, which states that writing zero bytes to a file has no effect.

4.5.2 Problems and Restrictions

This section describes DEC C RTL problems and restrictions.

4.5.2.1 SYS_ERRLIST and SYS_NERR Symbols (Alpha Only)

V6.2

Alpha

The symbols SYS_ERRLIST and SYS_NERR have been removed from object library SYS$LIBRARY:STARLET.OLB. If you see these symbols being undefined during a link operation, read file SYS$LIBRARY:DECC$SYS_ERRLIST.COM for instructions on how to resolve this problem. ♦

4.5.2.2 DECC$SHR.EXE (VAX Only)

V6.2

VAX

A new package of internationalization routines has been added to the DEC C RTL (see Section 4.5.1.3) and the transfer vector has been extended, necessitating an increase in the minor ident (GSMATCH) of the image. Because of this, programs using the DEC C RTL (DECC$SHR.EXE) linked on OpenVMS VAX Version 6.2 systems will not run on prior versions of OpenVMS VAX. ♦

4.5.2.3 Internationalization Compatibility Problem with DECwindows Motif

V6.2

Applications that call the Xlib locale routines in DECwindows Motif Version 1.2–3 using the method described in Section 4.18.5.3 of the *DECwindows Motif Version 1.2–3 for OpenVMS Release Notes* will continue to function on OpenVMS Version 6.2. However, because the locale support in Xlib is not compatible with the support in the DEC C Run-Time Library (DECC$SHR.EXE), Xlib does not use the locale environment provided by the C library. Therefore, setting the locale in the C library does not affect the behavior of DECwindows Motif, although it does affect C library routines such as strcoll( ). Setting the locale in Xlib changes the behavior of DECwindows Motif but does not affect C library routines.

Applications that call the Xlib locale routines in DECwindows Motif Version 1.2 using the method described in the *DECwindows Motif Version 1.2–3 for OpenVMS Release Notes*, Section 4.17.6.3, do *not* run correctly on OpenVMS Version 6.2 unless you install the DECwindows Motif Version 1.2 remedial kit appropriate for your system (see the table in Section 3.11.2.5). With the remedial kit installed, the Xlib locale routines work normally, but Xlib does not use the locale environment provided by the C library.

Digital expects to correct this problem in a future release of DECwindows Motif.

4.5.2.4 Regular Expressions

V6.2

Regular expressions in the source string passed to decc$to_vms do not work properly. Operations result in either an ACCVIO status return or a return value of 0 indicating that no files match the wildcard.

4.5.2.5 Simultaneously Opened Files Limited to 256

V6.2

DEC C RTL limits the number of simultaneously opened files to 256.

4.5.3 Corrections

This section describes corrections to the DEC C RTL.

4.5.3.1 Maximum Record Size Is Now Set for Stream Files

V6.2

DEC C RTL no longer ignores the Maximum Record Size (mrs) value when it is provided by the caller. Previously, stream files would always be created with a maximum record size of 32767, regardless of the specified mrs value. Note that the DEC C RTL does not validate the integer used as the mrs value. Application developers should not use values greater than 32767 to specify record sizes to RMS.

4.5.3.2 Undefined Record Formats

V6.2

The I/O system now properly inherits a record attribute of none when a previous version of the file had an undefined record format. Previously, the newly created file would be created with carriage return record attributes. The functions `fopen`, `open`, and `creat` have also been corrected to allow the RMS option "rat=none" to override the record attributes of the previous version of the file.

4.5.4 Programming Tips

This section contains helpful programming tips for DEC C RTL users.

4.5.4.1 RMS–11 Stream Format Files

V6.1

RMS defines an RMS–11 stream format file that corresponds to the value of FAB$C_STM for the record format. This format definition causes the RMS record operation SYS$GET to remove leading null bytes from each record. Because DEC C RTL processes this file type in record mode, it is not a suitable file format for binary data unless it is explicitly opened with "ctx=stm" to cause the raw bytes of data from the file to be returned.

4.6 DECmigrate: Translated Image Support (Alpha Only)

V6.1

Alpha

DECmigrate Version 1.1A runs on OpenVMS Alpha Version 6.1. The images it translates require this version or a later version to execute. Translated images are generally forward compatible but not backward compatible; that is, images translated with DECmigrate Version 1.1A can run only on OpenVMS Alpha Version 6.1 or later while images translated with DECmigrate Version 1.0 can run on OpenVMS Alpha Version 1.0 and later. Table 4–2 correlates the versions of OpenVMS Alpha systems with the different versions of DECmigrate that support them.

Table 4–2 Support for Translated Images on OpenVMS Alpha Versions

DECmigrate Version Used to Translate Images	OpenVMS Alpha Support for Translated Images		
	Version 1.0	Version 1.5	Version 6.1 and later
Version 1.0	Yes	Yes	Yes
Version 1.1	No	Yes	Yes
Version 1.1A	No	No	Yes ♦

4.7 DECthreads

The following sections contain release notes pertaining to DECthreads.

See the *Guide to DECthreads* for detailed information about using DECthreads software.

4.7.1 Problems and Restrictions

This section describes known DECthreads problems and restrictions.

4.7.1.1 Dynamic Image-Activation

V6.2

Applications that use thread-safe run-time libraries might not be able to use LIB$FIND_IMAGE_SYMBOL to dynamically activate DECthreads or products that depend on DECthreads.

Certain run-time libraries use conditional synchronization mechanisms. These mechanisms typically are enabled during image initialization when the run-time library is loaded only if the process is multithreaded. If the process is not multithreaded, the synchronization is disabled.

If a single-threaded application attempts to use LIB$FIND_IMAGE_SYMBOL to activate an image that would cause the process to become multithreaded, a LIB$_REENTRANCY error occurs and the application is terminated.

This termination ensures that threads can never be present when run-time libraries are not using the proper synchronization.

To work around this problem, link the image that calls LIB$FIND_IMAGE_SYMBOL against CMA$RTL.

4.7.1.2 Exit Handler Routine

V6.1

If you try to abort a program that uses DECthreads functions in an exit handler routine by using a Ctrl/Y sequence followed by the DCL EXIT command (or almost any DCL command), the program may hang indefinitely in the exit handler routine.

One instance of this problem occurs when you type a Ctrl/Y sequence to interrupt a multithreaded program in the middle of a C RTL I/O function. The problem is with the operating system, not with your program code or the C RTL code. To release your program from the exit handler routine, type another Ctrl/Y sequence followed by the DCL STOP command.

Digital expects to correct this problem in a future release of the OpenVMS operating system.

4.7.1.3 Routines Can Terminate Process (Alpha Only)

V1.0

Alpha

The DECthreads routines `cma_thread_exit_error`, `cma_thread_exit_normal`, and `pthread_exit` should terminate the calling thread only. However, these routines erroneously cause the process to terminate when they are called in the initial thread.

This problem will be corrected in a future version of OpenVMS Alpha. ♦

4.8 DECTPU for DECwindows Motif

The following sections contain release notes pertaining to DECTPU for DECwindows Motif.

4.8.1 Problems and Restrictions

This section describes DECTPU for DECwindows Motif problems and restrictions.

4.8.1.1 Small Display Monitors and DECwindows Motif Applications

V6.0

When running DECwindows Motif DECTPU on small display monitors, the main window can be less than fully visible.

To correct this condition, follow these steps:

1. Add the following resources to your DECTPU X resource file:

```
Tpu.Tpu$MainWindow.X:                    0
Tpu.Tpu$MainWindow.Y:                    0
Tpu.Tpu$Mainwindow.Rows:                 21
Tpu*condensedFont:                       on
Tpu*fontSetSelection:                    1
```

2. Copy the resource file from SYS$LIBRARY:EVE.DAT and add the previous lines.
3. Use logical name TPU$DEFAULTS to point at the new resource file.

   The following example invokes the EVE DECwindows Motif user interface using the X resource file named eve_small_window.dat in your login directory to edit the file LOGIN.COM.

```
$ DEFINE TPU$DEFAULTS SYS$LOGIN:EVE_SMALL_WINDOW.DAT
$ EDIT/TPU/INTER=DECWINDOWS LOGIN.COM
```

4.9 DECwindows Motif Transport

The following sections contain release notes pertaining to the DECwindows Motif Transport.

4.9.1 Corrections

This section describes corrections to the DECwindows Motif Transport.

4.9.1.1 User-Written Modifications

VAX V6.0
Alpha V6.1

The DECwindows Motif Transport has been modified to correct previous disconnect problems. Under certain conditions, pending ASTs could be sent after the Transport cleanup process had begun. As a result, the AST could attempt to access data that had already been deallocated, which caused access violations or system crashes.

Queuing the CLOSE_AND_DEALLOCATE_AST AST from the DECW$$TCPIP_CLOSE routine is an incomplete solution. The CLOSE_AND_DEALLOCATE_AST AST is queued to the current operation mode (EXEC), and this AST could be sent before pending USER mode ASTs.

The Transport Function Table (XTFT) data structure has been modified and a new common routine has been added. The XTFT data structure has been increased by one longword to provide an additional routine entry, XTFT$A_DISCONNECT. The new common routine is DECW$XPORT_DISCONNECT.

If you have written your own DECwindows Motif Transport, you can implement these modifications, based on the Sample Transport implementation Digital recommends in the *VMS DECwindows Transport Manual*.

Note

You must recompile any user-written specific transport that uses the XTFT$C_LENGTH constant.

- Create a new routine to remove the cleanup instructions from the CLOSE_AND_DEALLOCATE_AST AST. Store this new routine address in the XTFT data structure at offset XTFT$A_DISCONNECT. The XTFT is initialized in the DECW$TRANSPORT_INIT routine. This new routine takes the IXTCC parameter.
- From DECW$$TCPIP_CLOSE, queue CLOSE_AND_DEALLOCATE_AST as a USER mode AST.
- Inside CLOSE_AND_DEALLOCATE_AST, call the Common Transport routine DECW$XPORT_DISCONNECT, passing an IXTCC parameter. DECW$XPORT_DISCONNECT will call the routine stored in XTFT[XTFT$A_DISCONNECT]. By dispatching the disconnect routine through the Common Transport layer, the process will change to EXEC mode as required to deallocate data structures.

4.10 Delta/XDelta Debugger

The following sections contain release notes pertaining to the Delta/XDelta debugger.

4.10.1 Changes and Enhancements

This section describes changes or enhancements to the software.

4.10.1.1 Image Activation (Alpha Only)

V6.1

Alpha

Because of enhancements to the OpenVMS Debugger, the image activator on OpenVMS Alpha systems has been modified to automatically activate SYS$SHARE:SYS$SSISHR.EXE when an image is debugged using the RUN /DEBUG command or is linked using the /DEBUG qualifier.

If the Delta/XDelta debugger is being used, SYS$SHARE:SYS$SSISHR.EXE may be automatically activated for you. The presence of this image should not alter your program's correctness, but if your program is sensitive to virtual address layout or if for some reason SYS$SHARE:SYS$SSISHR.EXE is not installed properly on your system, you may want to bypass its automatic activation.

To keep the image activator from activating SYS$SHARE:SYS$SSISHR.EXE for you, define the logical name SSI$AUTO_ACTIVATE to be "OFF" before running the program to be debugged with Delta/XDelta. ♦

4.10.2 Documentation Changes and Corrections

This section describes corrections to the Delta/XDelta documentation.

4.10.2.1 *OpenVMS Delta/XDelta Debugger Manual*

V1.5 The base register default offset for OpenVMS Alpha is 10000_{16}, not 100000_{16}, as documented in the description of the ;X (Load Base Register) command.

4.11 Device Support on OpenVMS Alpha Systems

Alpha The following sections contain release notes pertaining to OpenVMS Alpha device drivers.

4.11.1 Changes and Enhancements (Version 6.2)

This section describes OpenVMS Alpha Version 6.2 device support changes or enhancements.

4.11.1.1 Driver Compatibility for Version 6.2

V6.2 Most OpenVMS Alpha Version 6.1 device drivers will run on OpenVMS Alpha Version 6.2 without recompiling and relinking. However, you must make source code changes and recompile and relink the following OpenVMS Alpha Version 6.1 device drivers:

- OpenVMS Alpha device drivers written for PCI, EISA, or ISA devices that perform direct memory access (DMA)
- OpenVMS Alpha SCSI port and class device drivers

For more information about the OpenVMS Alpha Version 6.2 changes required for device drivers written for PCI, EISA, or ISA devices that perform direct memory access (DMA), see Section 4.11.1.3. For more information about OpenVMS Alpha SCSI port and class drivers, see Section 4.11.1.5.

For details about ISA device configurations, see Appendix A.

4.11.1.2 Improving Performance of Device Drivers Written in C

V6.2 Additional pragmas have been added to the C header files that define the following data structures in SYS$LIBRARY:SYS$LIB_C.TLB:

ACB	AST Control Block
BOD	Buffer Object Descriptor
CCDB	Class Driver Data Block
CDRP	Class Driver Request Packet
CPU	CPU Database
CRAMH	CSR Register Access Mailbox Header
DDB	Device Data Block
DDT	Driver Dispatch Table
DPT	Driver Prologue Table
FDT	Function Decision Table
FDT_CONTEXT	Function Decision Table Contect Structure
FKB	Fork Block
FT_UCB	Pseudo Terminal Driver UCB
HW_CRAM	Hardware CSR Register Access Mailbox
INTSTK	Interrupt Stack Structures
IRPE	Interrupt Request Packet Extension

KPB	Kernel Process Block
LTRM_UCB	Logical Terminal UCB Extension
PCB	Process Control Block
PHD	Process Header
PTE	Page Table Entry
SB	System Block
SCDRP	SCSI Class Driver Reset Packet
SPDT	SCSI Port Descriptor Table
TAST	Terminal AST Block
TPD_UCB	Terminal Port Driver Dependent Extension
TQE	Timer Queue Entry
TTY_UCB	Terminal Driver UCB

The compiler generates more efficient code when you reference the members of these structures. To take advantage of this enhancement and improve the performance of your OpenVMS Alpha device driver that is written in C, recompile your driver.

4.11.1.3 PCI DMA Window Changes

V6.2

This release note should be read by anyone writing a driver for a PCI, EISA, or ISA device.

PCI bus implementations on Alpha platforms define two ways for PCI devices to access main memory: scatter/gather memory access and physical memory access. In **scatter/gather memory access**, the PCI address generated by a PCI device is translated to a main memory address by a scatter/gather table. In **physical memory access** the PCI address generated by a PCI device is translated to a main memory address by the addition of a constant. Scatter/gather memory access is called **scatter/gather DMA**. Physical memory access is called **physical (or direct) DMA**.

Each memory access technique has advantages and disadvantages. Scatter/gather DMA allows access to all system memory, but it is more complex to program. Physical DMA is easier to program, but it may limit the amount of main memory that can be addressed.

Alpha PCI platforms implement scatter/gather DMA and physical DMA through **DMA address windows** on the PCI bus. A PCI DMA address window is an address range on the PCI bus through which PCI devices (and EISA/ISA devices behind the PCI/EISA bridge or PCI/ISA bridge) access main memory. The Alpha platforms that support PCI buses (AlphaServer 2000, AlphaServer 2100, AlphaServer 1000, AlphaServer 400, AlphaServer 8200, AlphaServer 8400, AlphaStation 200, AlphaStation 250) provide a minimum of two DMA address windows on the PCI bus. A DMA address window can be a physical DMA window, where a PCI bus address is a linear function of a system memory address, or a scatter/gather DMA window, where a PCI bus address undergoes a page table translation before becoming a main memory address.

Alpha platforms that support PCI buses have both a physical DMA window and a scatter/gather DMA window. In OpenVMS Alpha Version 6.2, the scatter/gather DMA window is based at PCI address 0 and extends to a maximum address of 3FFFFFFF (the actual size of the scatter/gather DMA window is a function of the amount of physical memory in the system). In OpenVMS Alpha Version 6.2,

the physical DMA window is typically based at a PCI address above the scatter /gather DMA window.

To allow drivers to find the base of the physical DMA window in a platform independent manner, two new function codes have been added to the bus support routine IOC$NODE_DATA. The basic definition of the IOC$NODE_DATA routine is as follows:

```
int ioc$node_data (CRB  *crb, int  function_code, void *user_buffer)
```

Inputs:

crb	Address of CRB. IOC$NODE_DATA uses the crb$l_node field and the vec$ps_adp field to find the data structures associated with the I/O bus to which this device is connected.
function_code	From [lib.lis]iocdef.sdl. Specifies information to be returned by IOC$NODE_DATA.
user_buffer	Address of caller's buffer. On success, the requested information is returned in the caller's buffer.

Outputs:

SS$_NORMAL	Normal successful completion. Requested information is returned in the caller's buffer.
SS$_ILLIOFUN	Unrecognized function code.
SS$_BADPARAM	Bad parameter. Usually this means that crb$l_node contains an invalid slot number. Check that the driver has been loaded with the /node qualifier.

When called with function code IOC$K_DIRECT_DMA_BASE, the IOC$NODE_ DATA routine returns the base address of the physical DMA window. The base address is returned as a 64-bit value in anticipation of future 64-bit I/O buses. For this reason, the caller of the IOC$NODE_DATA routine should make sure the user_buffer argument points to a quadword cell when using the IOC$K_DIRECT_ DMA_BASE function code.

When called with function code IOC$K_DIRECT_DMA_SIZE, the IOC$NODE_ DATA routine returns the size of the physical DMA window, expressed in megabytes. The size of the direct DMA window is returned as a 32-bit value.

Using the physical DMA window for device DMA is straightforward. Once you have found the base of the physical DMA window using the IOC$NODE_DATA routine, you must adjust the DMA address that you assign to the device by adding the physical DMA window base to the main memory DMA buffer address. For example, on a typical system the physical DMA window is based at the PCI address 40000000 and extends to 7FFFFFFF. PCI addresses in the range from 40000000 to 7FFFFFFF are passed to main memory addresses 0 to 3FFFFFFF. This means that if you have a DMA buffer at main memory address 0, the PCI device would access this buffer at PCI address 40000000. The correspondence of main memory address and PCI DMA addresses is derived by the following formula:

```
PCI DMA address = (main memory buffer address) + (base of physical DMA window)
```

The main reason for locating the PCI DMA windows this way is to support bit-limited ISA devices. In OpenVMS Alpha Version 6.1, bit-limited ISA bus masters cannot use the scatter/gather DMA window because ISA bus masters do not have enough address bits to generate an address above 16 MB. For this reason, the Scatter/Gather window was moved to PCI address 0. The scatter/gather table is

now managed such that EISA and ISA devices that perform DMA in the scatter /gather window always get bus addresses below 16 MB.

Note that the size of the physical DMA window limits the amount of physical memory that can be addressed by a PCI, EISA, or ISA device. In the physical DMA window, it is not possible for an I/O device to address more than 1 GB of physical memory. For this reason, on large memory systems you may want to code your driver to perform DMA in the scatter/gather window. To perform scatter/gather DMA, use the standard counted resource management routines described in the *OpenVMS AXP Device Support: Developer's Guide*.

4.11.1.4 Device IPL Setup for Drivers

V6.2

Alpha hardware platforms that support PCI, EISA, and ISA buses deliver I/O device interrupts at different IPLs, either 20 or 21. The IPL at which device interrupts are delivered can change if you move the device from one platform to another. This is a problem if the driver declares its device IPL to be 20, and then that driver is executed on a machine that delivers I/O device interrupts at IPL 21.

The simplest solution to this problem is for PCI, EISA, and ISA device drivers to use IPL 21. This will work correctly on platforms that deliver I/O device interrupts at IPL 20 and on platforms that deliver I/O device interrupts at IPL 21.

A future release of OpenVMS Alpha may provide a platform independent mechanism for drivers to determine the device IPL dynamically.

4.11.1.5 SCSI Port and Class Driver Changes

V6.2

The following sections describe changes to OpenVMS Alpha SCSI class and port device drivers.

4.11.1.5.1 OpenVMS Alpha SCSI Class Driver Changes

V6.2

Some SCSI-2 port interface (SPI) macros that supported OpenVMS Alpha SCSI class device drivers for Version 6.1 have been changed for OpenVMS Alpha Version 6.2. Table 4–3 lists the obsolete Version 6.1 macros and the Version 6.2 macros that have similar functions.

Note that many of the Version 6.2 SPI macros use different names, and most of them require new arguments. For this reason, do not replace each reference to an obsolete SPI macro without taking into account the OpenVMS Alpha Version 6.2 interface changes.

You can obtain the new SPI$ interface macro definitions by extracting them from SYS$LIBRARY:LIB.MLB, using the following command:

```
$ LIBRARY/EXTRACT=SPI$*/OUTPUT=SPI$DEFINITIONS.MAR SYS$LIBRARY:LIB.MLB
```

Table 4–3 Obsolete SPI Interface Macros

Obsolete V6.1 Macro	Similar Version 6.2 Macro
SPI$MAP_BUFFER	SPI$BUFFER_MAP
SPI$UNMAP_BUFFER	SPI$BUFFER_UNMAP
SPI$SEND_COMMAND	SPI$SEND_COMMAND

(continued on next page)

Table 4–3 (Cont.) Obsolete SPI Interface Macros

Obsolete V6.1 Macro	Similar Version 6.2 Macro
SPI$SET_CONNECTION_CHAR	SPI$CONNECTION_CHAR_SET
SPI$GET_CONNECTION_CHAR	SPI$CONNECTION_CHAR_GET
SPI$RESET	SPI$RESET_SCSI_BUS
SPI$ABORT_COMMAND	SPI$ABORT_COMMAND
SPI$DISCONNECT	SPI$DISCONNECT
SPI$ALLOCATE_COMMAND_BUFFER	SPI$CMD_BUFFER_ALLOC
SPI$DEALLOCATE_COMMAND_BUFFER	SPI$CMD_BUFFER_DEALLOC
SPI$RESERVED1	-
SPI$RESERVED2	-
SPI$CONNECT	SPI$CONNECT
SPI$FREEZE_QUEUE	SPI$QUEUE_FREEZE
SPI$TQE_WAIT	Use EXE$KP_TQE_WAIT
SPI$RELEASE_QUEUE	SPI$QUEUE_RELEASE
SPI$FLUSH_QUEUE	SPI$QUEUE_FLUSH
-	SPI$RESET_DEVICE

4.11.1.5.2 OpenVMS Alpha SCSI Tagged Command Queuing

V6.2 For OpenVMS Alpha Version 6.2, the OpenVMS Alpha SCSI port device driver interface has been changed to support the tagged command queuing architecture of the SCSI-2 standard as described in specification ANSI X3.131-1994.

4.11.1.5.3 Enhanced SCSI Interface to IO$_DIAGNOSE Function

V6.2 The $QIO IO$_DIAGNOSE function has been enhanced for the following SCSI class drivers: GKDRIVER, DKDRIVER, and MKDRIVER. The function now allows for the return of autosense data into a user-specified buffer. In addition, a new data structure, called the S2DGB (SCSI-2 Diagnose Buffer), has been placed in STARLET to facilitate the use of the IO$_DIAGNOSE function.

The new $QIO IO$_DIAGNOSE parameters are as follows:

Parameter	Use
P1	S2DGB base address
P2	S2DGB length
P3	Reserved, should be zero
P4	Reserved, should be zero
P5	Reserved, should be zero
P6	Reserved, should be zero

Example 4–1 shows the only S2DGB format that can be accepted.

Note that this structure is upwardly compatible with the existing structure used by the IO$_DIAGNOSE function.

Example 4–1 OpenVMS SCSI-2 Diagnose Buffer (S2DGB)

```
+-------------------------------------------------+
|              S2DGB$L_OPCODE                     | :00
+-------------------------------------------------+
|              S2DGB$L_FLAGS                      | :04
+-------------------------------------------------+
|              S2DGB$L_32CDBADDR                  | :08
+-------------------------------------------------+
|              S2DGB$L_32CDBLEN                   | :0C
+-------------------------------------------------+
|              S2DGB$L_32DATADDR                  | :10
+-------------------------------------------------+
|              S2DGB$L_32DATLEN                   | :14
+-------------------------------------------------+
|              S2DGB$L_32PADCNT                   | :18
+-------------------------------------------------+
|              S2DGB$L_32PHSTMO                   | :1C
+-------------------------------------------------+
|              S2DGB$L_32DSCTMO                   | :20
+-------------------------------------------------+
|              S2DGB$L_32SENSEADDR                | :24
+-------------------------------------------------+
|              S2DGB$L_32SENSELEN                 | :28
+-------------------------------------------------+
|                                                 | :2C
+--                                             --+
|              Reserved                           | :30
+--                                             --+
|              Should Be Zero                     | :34
+--                                             --+
|                                                 | :38
+-------------------------------------------------+
```

The correct length of the structure is defined by the constant S2DGB$K_XCDB32_LENGTH (value: 60-decimal). The addresses in this structure are 32-bit virtual addresses. The fields in the structure are as follows:

S2DGB$L_OPCODE

This field should contain S2DGB$K_OP_XCDB32 (value: 1).

S2DGB$L_FLAGS

This field should contain the bit fields shown in the following table. Note that these bit definitions start at bit 0 and omit no bits. This is required for compatibility with the existing IO$_DIAGNOSE interface.

S2DGB$V_READ	This bit should be 1 if the operation being performed is a read. If the operation is a write, this bit should be 0.
S2DGB$V_DISCPRIV	This bit should contain the DiscPriv bit value to be used in the IDENTIFY message sent with this operation. If S2DGB$V_TAGGED_REQ is 1, then this bit should be ignored. Note that this bit may be ignored by some ports.
S2DGB$V_SYNCHRONOUS	This bit is ignored since its value is beyond the control of the user in SCSI-2 drivers.
S2DGB$V_OBSOLETE1	This bit is ignored. Formerly it represented the disabling of command retries, which is now beyond the control of the user in SCSI-2 drivers.
S2DGB$V_TAGGED_REQ	When this bit is 1, the operation will be processed as using tagged command queuing and S2DGB$V_ TAG should define the tag value to be used. When this bit is 0, the operation will be processed without benefit of tagged command queuing. Ports that do not support tagged command queuing always behave as if this bit is 0. Note that some ports simulate untagged operation using appropriately tagged operations.
S2DGB$V_TAG	If S2DGB$V_TAGGED_REQ is 1, then this 3-bit field should contain one of the following coded constant values: S2DGB$K_SIMPLE indicates that the command is to be sent with the SIMPLE queue tag. S2DGB$K_ORDERED indicates that the command is to be sent with the ORDERED queue tag. S2DGB$K_EXPRESS indicates that the command is to be sent with the HEAD OF QUEUE queue tag. If S2DGB$V_TAGGED_REQ is 0, then this field is ignored. Ports that do not support tagged command queuing always ignore the S2DGB$V_TAG field and send all commands as untagged operations. Note that automatic contingent allegiance processing is not accessible through the IO$_DIAGNOSE function.

S2DGB$V_AUTOSENSE	When this bit is 1, S2DGB$L_32SENSEADDR and S2DGB$L_32SENSELEN should contain a valid sense buffer address and length. If a CHECK CONDITION or COMMAND TERMINATED status is returned, REQUEST SENSE data will be returned in the buffer defined by S2DGB$L_ 32SENSEADDR and S2DGB$L_32SENSELEN. When this bit is 0, the buffer described by S2DGB$L_32SENSEADDR and S2DGB$L_ 32SENSELEN is ignored and sense data will be returned only if the CDB defines a REQUEST SENSE command. When S2DGB$V_AUTOSENSE is zero, the class driver saves the autosense data in pool and returns it to the next IO$_DIAGNOSE, if that IO$_DIAGNOSE has a REQUEST SENSE CDB. All other bits in S2DGB$L_FLAGS should be zero.

S2DGB$L_32CDBADDR

This field should contain the 32-bit virtual address of the SCSI Command Data Block (CDB) to be sent to the target by this IO$_DIAGNOSE operation.

S2DGB$L_32CDBLEN

This field should contain the number of bytes in the SCSI Command Data Block (CDB) to be sent to the target by this IO$_DIAGNOSE operation. (Legal values: 2 to 248. However, some ports may restrict CDBs to smaller lengths. Recommended values: 2 to 16.)

S2DGB$L_32DATADDR

This field should contain the 32-bit virtual address of the DATAIN or DATAOUT buffer to be used with this SCSI operation. If the CDB being sent to the target does not use a DATAIN or DATAOUT buffer, then this field should be zero.

S2DGB$L_32DATLEN

This field should contain the number of bytes in the DATAIN or DATAOUT buffer associated with this operation. If the CDB being sent to the target does not use a DATAIN or DATAOUT buffer, then this field should be zero. (Legal values: 0 to UCB$L_MAXBCNT. Recommended values: 0 to 65,536. All ports are required to support at least 65,536 byte data transfers.)

S2DGB$L_32PADCNT

This field should contain the number of padding DATAIN or DATAOUT bytes required by this operation. (Legal values: 0 to the maximum number of bytes in a disk block on this system minus one. Current legal values: 0 to 511.)

S2DGB$L_32PHSTMO

This field should contain the number of seconds that the port driver should wait for a phase transition to occur or for delivery of an expected interrupt. If S2DGB$V_TAGGED_REQ is 1 or this field contains a 0 or 1, then the current phase transition timeout setting will not be changed. (Legal values: 0 to 300 {5 minutes}.)

S2DGB$L_32DSCTMO

This field should contain the number of seconds that the port driver should wait for a disconnected transaction to reconnect. If S2DGB$V_TAGGED_REQ is 1 or this field contains a 0 or 1, then the current disconnect timeout setting will not be changed. (Legal values: 0 to 65,535 {about 18 hours}.)

S2DGB$L_32SENSEADDR

If S2DGB$V_AUTOSENSE is 1, then this field should contain the 32-bit virtual address of the sense buffer to be used by this SCSI operation. If S2DGB$V_ AUTOSENSE is 0, this field will be ignored.

S2DGB$L_32SENSELEN

If S2DGB$V_AUTOSENSE is 1, then this field should contain the number of bytes in the sense buffer associated with this operation. (Legal values: 0 to 255. Note: a value of 0 instructs the class driver to discard any sense data received. Recommended value: 18. Some ports may restrict the number of sense bytes to 18.) If S2DGB$V_AUTOSENSE is 0, this field will be ignored.

If all parameters are valid, the class driver will invoke the necessary port functions to send the CDB, transfer the data, and return, save or discard sense data as defined by the input S2DGB. Upon completion, the return IOSB will have the following format:

```
+----------------------------+----------------------------+
|      byte count <15:0>     |       port VMS status      | :00
+------------+---------------+----------------------------+
| SCSI status |     zero     |      byte count <31:16>    | :04
+------------+---------------+----------------------------+
```

The DKDRIVER, GKDRIVER, and MKDRIVER class drivers, which implement other QIO functions, might intermix other tagged requests with IO$_DIAGNOSE requests. The order in which requests are sent generally match the order in which requests are presented to the driver. An exception to this ordering occurs when the driver receives REQUEST SENSE for which autosense data previously has been recovered and stored. In this case, the IO$_DIAGNOSE will complete immediately and no command will be sent to the target.

The DKDRIVER, GKDRIVER, and MKDRIVER class drivers permit only one IO$_DIAGNOSE operation to be active (in the start I/O routine) at given time, except as described in the next paragraph. However, applications must single thread IO$_DIAGNOSE requests in order to properly detect the presence of sense data and send the required REQUEST SENSE command. This is consistent with the VAX IO$_DIAGNOSE behavior. For example, if three reads are issued with no waiting and the first read gets a CHECK CONDITION, the sense data will be discarded by the target when the second read arrives.

The DKDRIVER, GKDRIVER, and MKDRIVER drivers permit more than one IO$_DIAGNOSE operation to be active (in the start I/O routine) only when all active operations have the S2DGB$V_AUTOSENSE flag equal to 1. Upon encountering the first IO$_DIAGNOSE with S2DGB$V_AUTOSENSE equal to 0, the class driver will apply the restrictions described in the previous paragraph.

4.11.1.5.4 GKDRIVER Command Queuing Enhancement

V6.2

The GKDRIVER device driver supports the additional IO function code IO$_ SETCHAR, which allows an application to request that a connection be characterized as tagged-command-queuing capable. By default, GKDRIVER characterizes connections as nonqueuing.

To implement the IO$_SETCHAR function, do the following:

1. Call IO$_SETCHAR with the P1 parameter equal to zero or one.
2. If the caller passes a P1 value of zero, then the connection characteristics will be set so as to disable queuing behavior.

3. If the caller passes a value of one and the port supports tagged command queuing, then SPI$V_CC_CMDQ will be set to one to indicate queuing. Also, SPI$V_CC_SCSI_2 (indicating SCSI-2 characteristics) is set. Note that callers are responsible for requesting queuing only to those devices that have hardware queuing support.
4. If the caller passes a value of one and the port does not support queuing, then the connection characteristics shall continue to reflect nonqueuing characteristics and success shall be returned, thereby allowing the caller's application to run transparently on both queuing and nonqueuing platforms.

Note

If the connection is characterized as a queuing connection, then disconnect status may not be changed, and disconnects will be enabled by default. If the connection does not support queuing, then connection characteristics may be modified (using IO$_DIAGNOSE) to either enable or disable disconnects as the caller requests in the S2DGB.

4.11.2 Problems and Restrictions (Version 6.2)

This section describes known problems and restrictions related to OpenVMS Alpha Version 6.2 device support.

4.11.2.1 Hardware Support for Multiple-Host SCSI VMScluster Systems

V6.2

This note describes supported hardware requirements for multiple-host SCSI VMScluster systems, and it explains how to determine if your hardware and firmware meet minimum version requirements.

4.11.2.1.1 Supported Hardware for SCSI VMScluster Systems

V6.2

Table 4–4 shows the supported hardware components for SCSI VMScluster systems; it also lists the minimum required revision for these hardware components (that is, for any component, you must use either the version listed in Table 4–4 or a subsequent version).

Table 4–4 Supported Hardware for SCSI VMScluster Systems

Component	Supported Item	Minimum Version or H/W Revision	How to Find Your Version
Hosts	AlphaServer 400 4/xxx	See footnote[1]	Console SHOW VERSION command
	AlphaServer 1000 4/xxx	See footnote[1]	
	AlphaServer 2000 4/xxx	See footnote[1]	
	AlphaServer 2100 4/xxx	See footnote[1]	
	AlphaStation 200 4/xxx	See footnote[1]	
	AlphaStation 250 4/xxx	See footnote[1]	
	AlphaStation 400 4/xxx	See footnote[1]	

[1]The minimum revision of this component for SCSI VMScluster configurations is the version included in the Version 3.2 Firmware Kit, on the May, 1995 CD–ROM. The revision number is listed in the *Firmware Release Notes Overview* that accompanies that Firmware Kit.

(continued on next page)

Table 4–4 (Cont.) Supported Hardware for SCSI VMScluster Systems

Component	Supported Item	Minimum Version or H/W Revision	How to Find Your Version
Disks	RZ26	N/A[2]	Console SHOW DEVICE command
	RZ26L	442D	
	RZ28	442D	
	RZ28B	0006	
	RZ29B	0006	
Controller	HSZ40	2.5	Console SHOW DEVICE command
Bus Isolators	DWZZA–AA	E01	Examine product sticker
	DWZZA–VA	F01	
Adapters[3]	Integral system adapter	N/A	
	KZPAA (PCI to SCSI)		

[2]The version information for this component is incorrectly listed in Table 7-2 in the *OpenVMS Version 6.2 New Features Manual* as 392A. There is no version requirement for this component.

[3]You can configure other types of SCSI adapters in a system for single-host access to local storage.

4.11.2.1.2 Minimum SCSI Firmware Requirements for SCSI VMSclusters

V6.2 If your system contains RZ26L or RZ28 disks, their firmware revision level must be at or above 442D for use in a SCSI VMScluster.

Important Note

Only certain RZ26L and RZ28 firmware revisions can be safely upgraded to firmware revision level 442D. Refer to Section 4.11.2.1.3, to determine if your disks are capable of being upgraded to firmware revision level 442D. If your disk is capable of supporting firmware revision level 442D, use the RZTOOLS Utility that is described in Section 4.11.2.1.4 to update the disk's firmware.

4.11.2.1.3 Firmware Revision Level 442D Requirements

V6.2 Only the following combinations of disk drives and firmware revision levels are capable of being upgraded safely to firmware revision level 442D. Performing the update procedure on any other combination can permanently damage the disk.

Table 4–5 Revision Level 442D Firmware Compatibility

Disk Drive	Firmware Revision	Disk Filename
RZ26L	440C	RZ26L_442D_DEC.FUP
RZ28	441C or D41C	RZ28_442D_DEC2104.FUP

4.11.2.1.4 442D Firmware Installation Procedure

V6.2 If you determine that your disk requires revision level 442D firmware, and it is capable of being upgraded safely, use the following procedure to update the firmware. (See Table 4–5 for the filename of the disk you are upgrading.)

```
$ MCR SYS$ETC:RZTOOLS_ALPHA DKB500 /LOAD=SYS$ETC:disk filename
  Read in 262144 bytes.
  Current FW version - X440C
  Upgrading to       - DEC0
  Loading code  ......
  New code has been sent to the drive.
```

4.11.2.1.5 Documentation Correction for Supported Hardware for SCSI VMScluster Systems

V6.2 The version information for RZ26 disks is incorrectly listed in Table 7-2 in the *OpenVMS Version 6.2 New Features Manual* as 392A. The information in Section 4.11.2.1.1 supersedes this information. There is no version requirement for this component.

4.11.2.2 Unsupported Ethernet ISA Bus Adapters

V6.2 OpenVMS Alpha Version 6.2 does not support the DE203, DE204, and DE205 Digital Ethernet ISA bus adapters. Support for these devices will be available in the next release of the operating system. A driver update kit that supports these devices can be obtained by contacting your Digital support representative.

4.11.2.3 SCSI Port and Class Drivers

V6.2 The following sections describe OpenVMS Alpha SCSI class and port device driver restrictions.

4.11.2.3.1 Mounting Third-Party SCSI Disks

V6.2 Users of the Digital RZ25M, Imperial MegaRAM MG-SCSI2, Seagate ST31200N, and ST31250N disks as well as other disks not qualified by Digital may see errors logged against these disks when they are mounted for the first time after booting in OpenVMS Alpha V6.2. Although these disks mounted error-free in OpenVMS Alpha Version 6.1, users might see 1 to 11 errors in OpenVMS Alpha Version 6.2 on the first mount.

The errors logged against these disks do not indicate a real problem. They occur because a 10-byte mode sense command is now issued to a disk instead of a 6-byte mode sense. This approach will allow OpenVMS to work with a wider variety of third-party and other previously untried disks.

Digital is working with the parties responsible for maintaining the disk's firmware to correct this condition.

4.11.2.3.2 Fatal Drive Errors for TZ86, TZ87, and TZ88 SCSI Tape Drives

V6.2

When initializing a TZ86, TZ87, or TZ88 SCSI tape drive, a fatal drive error may occur. If the fatal drive error is accompanied by an error log entry of type "Extended Sense Data Received" and its associated sense data is labeled "Recovered Error, Error Log Overflow," contact your Digital support representative for a patch.

4.11.2.3.3 Add-On SCSI Adapters

V6.2

OpenVMS Alpha 6.2 supports various add-on SCSI adapters. Digital's AlphaGeneration platforms typically support one or more integral SCSI adapters, with the option of installing additional add-on SCSI adapters. Due to differences in device-naming conventions used between the Alpha console and OpenVMS, the OpenVMS device name may not match the name displayed by the console.

For example, the console designation for a SCSI device on the integral SCSI adapter may be DKA100. However, when two additional add-on SCSI adapters are added, the "A" designation becomes "C"; and DKA100 appears as DKC100 when OpenVMS is running.

Note that although the console and OpenVMS device names may be different, the unique specification of a device name from the console to the device name under OpenVMS will stay consistent, as long as add-on SCSI adapters are not added or removed.

4.11.2.3.4 HSZ40 and Transportable SCSI Disk Shadow-Set Members

V6.2

An HSZ40 Raid-Array Controller provides the capability of an OpenVMS initialized SCSI disk (that is, one with a Files–11 ODS-2 format on it), to be moved between an OpenVMS controlled SCSI bus and an HSZ40 controlled SCSI bus, without reinitializing the disk and losing data. Disks that contain this functionality are called **transportable** disks.

A SCSI disk initialized by the HSZ40 and then subsequently initialized by OpenVMS is called a **nontransportable** disk, and cannot be moved to an OpenVMS controlled SCSI bus without losing data.

OpenVMS Volume Shadowing requires that a SCSI disk supports the SCSI commands READ_LONG/WRITE_LONG. These SCSI commands in conjunction with OpenVMS Volume Shadowing are used to handle certain classes of errors as seen under normal volume shadowing operations. SCSI disks that support the READ_LONG/WRITE_LONG capability while connected to an OpenVMS controlled SCSI bus, lose this capability when the transportable disks are moved to the SCSI bus controlled by an HSZ40.

The lack of READ_LONG/WRITE_LONG capability is detected at shadow-set MOUNT time, by the following error:

```
MOUN$_DEVNOFE, device does not support FORCED ERROR handling
```

To correct this problem, specify the MOUNT qualifier /OVERRIDE=NO_FORCED_ERROR at shadow-set MOUNT time.

Note that specifying this MOUNT qualifier may cause shadow-set member SCSI disks to be removed from a shadow set if certain error conditions arise that cannot be corrected.

Digital recommends that HSZ40 nontransportable SCSI disks be used to contain shadow-set members that support READ_LONG/WRITE_LONG functionality, and offer benefits provided by the level of RAID chosen at initialization time.

4.11.2.3.5 KZTSA Adapter Restriction

V6.2

When using wide SCSI devices, a hardware restriction with the KZTSA TURBOchannel to the SCSI adapter is present in certain revisions of the KZTSA option.

When installing a KZTSA option, you must follow these guidelines:

1. Version A10_1 or later of the KZTSA firmware must be used.
2. If the KZTSA option revision is "J" or less, then the WIDE parameter must be disabled for each wide device using the KZTSA DEVPARAMS utility.
3. If the KZTSA option revision is "K" or greater, then the WIDE parameter may be kept enabled (default) for each device.

   For instructions about using the KZTSA DEVPARAMS utility, refer to the *KZTSA Users Guide*.

Note

These guidelines pertain only to wide SCSI devices used on the KZTSA. Narrow SCSI devices are not affected. Narrow SCSI devices may have all parameters enabled (default).

4.11.2.3.6 KZPSA SCSI Adapter Bus Resets

V6.2

On PCI-based AlphaServer systems with heavy I/O rates, KZPSA SCSI adapters may occasionally reset the SCSI bus. These bus reset events, which may appear in the system error log, indicate that a recoverable SCSI bus failure has occurred. They do not require field service notification.

The following display shows partial DIAGNOSE example output containing bus reset error log information.

```
VMS SCSI Error Type  5. Extended Sense Data from Device
                    .
                    .
SCSI Status              x02  Check Condition
                    .
                    .
                         Power On, Reset, or Bus Device Reset
                         Occurred
```

Digital expects to fix this problem in a future release.

4.11.2.3.7 SCSI Disk I/O Performance Degradation for KZMSA XMI and Adaptec 1742A Adapters

V6.2

As a result of the SCSI-2 Tagged Command Queuing (TCQ) support in OpenVMS Alpha Version 6.2, Digital has determined that customers with KZMSA XMI to SCSI and Adaptec 1742A adapters might experience a 20% SCSI disk I/O performance degradation because TCQ is not implemented for these adapters. The performance degradation is in the area of increased CPU cost per I/O. Customers running at less than maximum CPU utilization under OpenVMS Alpha Version 6.1 might not experience any degradation under OpenVMS Alpha Version 6.2.

Digital does not expect this situation to significantly affect DEC 7000 customers planning to upgrade to DEC 8000 Family systems using KZMSA adapters because the speed of those processors should offset the performance degradation. However, DEC 7000 customers who upgrade to OpenVMS Alpha Version 6.2 will experience the SCSI I/O disk performance degradation.

Digital expects that this will significantly affect existing DEC 2000 Model 300 systems customers that use the Adaptec 1742A SCSI adapter.

4.11.3 Changes and Enhancements (Version 6.1)

This section describes OpenVMS Alpha Version 6.1 device support changes or enhancements, which also apply to Version 6.2.

4.11.3.1 New Device Driver Interface

V6.1

Starting with Version 6.1, OpenVMS Alpha supports user-written device drivers and a new device driver interface known as the Step 2 driver interface, which replaces the temporary Step 1 driver interface provided in OpenVMS Alpha Versions 1.0 and 1.5. The Step 2 driver interface provides support for writing device drivers in C or another high-level language and that can conform to the OpenVMS Alpha calling standard. An example device driver written in C is provided in the SYS$EXAMPLES directory.

The following Alpha device support manuals are available for OpenVMS Alpha Version 6.1:

- *Creating an OpenVMS AXP Step 2 Device Driver from a Step 1 Device Driver*

  This manual describes how to convert an OpenVMS Alpha Step 1 device driver, written in VAX MACRO, to an OpenVMS Alpha Step 2 device driver, also written in VAX MACRO.

- *Creating an OpenVMS AXP Step 2 Device Driver from an OpenVMS VAX Device Driver*

  This manual describes how to convert an OpenVMS VAX device driver, written in VAX MACRO, to an OpenVMS Alpha Step 2 device driver, also written in VAX MACRO.

- *OpenVMS AXP Device Support: Developer's Guide*

  This manual describes how to write an OpenVMS Alpha device driver in a high-level language.

- *OpenVMS AXP Device Support: Reference*

  This manual provides reference material for creating OpenVMS Alpha device drivers, and describes the macros, system routines, and entry points used in converting OpenVMS VAX and Step 1 OpenVMS Alpha device drivers to Step 2 OpenVMS Alpha device drivers.

4.11.3.2 Step 1 Drivers Are Obsolete

V6.1

OpenVMS Alpha Version 6.1 does not support Step 1 driver interfaces, and source changes are required to convert a Step 1 driver to a Step 2 driver. For detailed information about how to convert a Step 1 driver to a Step 2 driver, see *Creating an OpenVMS AXP Step 2 Device Driver from a Step 1 Device Driver* and *OpenVMS AXP Device Support: Reference*.

If you attempt to load a Step 1 driver that was compiled and linked on any previous version of the OpenVMS Alpha operating system, the System Management utility (SYSMAN) will issue the following warning message, and the driver will not be loaded:

```
$ mcr sysman io connect mya0/noadapter/driver=$users:[jones]mydriver_v15
%SYSMAN-I-NODERR, error returned from node BEAMME
-SYSTEM-W-SYSVERDIF, system version mismatch; please relink
```

If you attempt to relink a Step 1 driver object module that was compiled on any previous version of the OpenVMS Alpha operating system, the linker will issue warning messages about undefined symbols for any references to the obsolete Step 1 support routines. The resultant image file is not loadable.

If you attempt to recompile a Step 1 driver module, the MACRO-32 Compiler will issue many compilation warnings and errors, including the following messages:

```
%AMAC-E-GENERROR, (1) generated ERROR: 0 DPTAB must declare driver STEP=2 ;

%AMAC-E-GENERROR, (1) generated ERROR: 0 FUNCTAB is an obsolete macro used
by STEP=1 drivers;
```

No object module will be produced.

4.11.3.3 OpenVMS Alpha System-Code Debugger

V6.1

OpenVMS Alpha Version 6.1 supports a new programming tool that can be used to debug nonpageable system code and device drivers running at any IPL. The OpenVMS Alpha System-Code Debugger lets you use the OpenVMS Debugger interface to observe and manipulate system code interactively as it executes.

For more information about how to use the system-code debugger, see the *OpenVMS AXP Device Support: Developer's Guide*. Before using the system-code debugger, note the problems and limitations described in Section 4.11.4.2.

4.11.4 Problems and Restrictions (Version 6.1)

This section describes known problems and restrictions related to OpenVMS Alpha Version 6.1 and later device support.

4.11.4.1 C Structure Member Names: Passing to Nested C Macros

V6.1

The file SYS$LIBRARY:SYS$LIB_C.TLB is a library of C header files that define structures corresponding to the structures defined for MACRO-32 in SYS$LIBRARY:LIB.MLB. These C structure definitions also include simple macros, which allow the use of the same field names in drivers written in C as those used in drivers written in MACRO-32 and BLISS.

A compile-time error results if both of the following occur:

- You pass one of these simple macro names for a structure member as part of a parameter to a C macro
- This macro passes this parameter to another macro

You can avoid this problem by using one of the following methods:

- "Flattening" the macro definition (by expanding the nested macros)
- Using C inline functions to replace the nested macros while maintaining the desired modularity
- Introducing additional temporary storage in the outer macro to avoid passing the input arguments directly to the nested macros

4.11.4.2 OpenVMS Alpha System-Code Debugger

V6.1

The OpenVMS Alpha System-Code Debugger has the following problems and limitations:

- The CALL command is not currently implemented.
- If you use the SET WATCH command, it uses nonstatic watchpoints. Because nonstatic watchpoints are slower than static watchpoints, their use is not recommended.
- With the OpenVMS Debugger, if you use the SET BREAK/EXCEPTION command, when the program hits an exception, the user could fix the problem and continue the program. With the system-code debugger, the SET BREAK/EXCEPTION command catches the problem later (just before a bugcheck). Therefore, it is not possible to fix the problem and proceed. The GO command will let the system bugcheck and write the crash dump.
- Setting breakpoints at IF statements in the C programming language does not always work. For example, if you have code like the following and you set a breakpoint at the first line, the system-code debugger may or may not hit the breakpoint (depending on the condition):

```
if (x)
{
        ....
} else
{
          ...
}
```

  To avoid this problem, you can set two breakpoints, one in the THEN clause and one in the ELSE clause, or set a breakpoint before the IF statement and then single-step until you reach the IF statement.

- When using the SET MODULE command on some images with the system-code debugger, you may receive an error message similar to the following:

```
%DEBUG-E-INVPD, procedure descriptor at 00028840 is not valid.
```

  This procedure descriptor is either in the init section for the image or in a paged-out image section. You can determine this by searching for the above value in the MAP file for the image to which the module belongs.

  If the value is not within the init image section or a pageable section, file a Problem Report and provide the error message along with the MAP file for the image. ♦

4.12 Device Support on OpenVMS VAX Systems

VAX

The following sections contain release notes pertaining to OpenVMS VAX device support.

4.12.1 Documentation Changes and Corrections

This section describes corrections to device support documentation.

4.12.1.1 *OpenVMS VAX Device Support Manual*

V6.1

The following sections describe corrections to the *OpenVMS VAX Device Support Manual*.

4.12.1.1.1 Linking a Device Driver

V6.1 Chapter 12 of the *OpenVMS VAX Device Support Manual*, Version 6.0, describes how to assemble, link, and load a device driver. In Step 3 of the procedure for preparing a driver for loading into the operating system, append the following text to the end of the procedure (following the paragraph that begins: "The resulting image must . . . "):

To produce an image with a symbol table compatible with the System Dump Analyzer (SDA), you must link again; this time, using the UNIVERSAL=* option statement (to include all global symbols and to ensure proper state of the REL bits in the object records). Relink as shown in the following example:

```
$ LINK /NOEXECUTABLE/NOTRACEBACK/NOSYSSHR -
_$     /SYMBOLS=MYDRIVER.EXE,-
_$     /SHARE=DUMMY_FILE_NAME,-
_$     /NOMAP,MYDRIVER1.OBJ,MYDRIVER2.OBJ,-
_$     SYS.STB/SELECTIVE,-
_$     SYS$INPUT/OPTION
_$     BASE=0
_$     UNIVERSAL=*
```

For more information about the Linker, see the *OpenVMS Linker Utility Manual*.

4.12.1.1.2 Device-Register I/O Space: Usage Restrictions

V6.1 Chapter 5 of the *OpenVMS VAX Device Support Manual*, Version 6.0, describes device driver coding and the restrictions on the use of device-register I/O space. The third sentence of the fifth bulleted paragraph in Section 5.2 states that the instructions that refer to UNIBUS adapter registers must use longword context. This is the wrong bus. The sentence should read:

> "Instructions that refer to MASSBUS adapter registers must use the longword context."

4.12.1.2 *OpenVMS VAX Device Support Reference Manual*

V6.1 The following sections describe corrections to the *OpenVMS VAX Device Support Reference Manual*.

4.12.1.2.1 COM$DRVDEALMEM Routine Synchronization

V6.1 Chapter 3 of the *OpenVMS VAX Device Support Reference Manual*, Version 6.0, contains a section describing the COM$DRVDEALMEM routine.

At the end of the paragraph under *Synchronization*, add the following sentence:

"If called at IPL$_SYNCH or higher, the routine executes the fork process."

4.12.1.2.2 CRB Data Structure

V6.1 Chapter 1, Section 1.7, of the *OpenVMS VAX Device Support Reference Manual*, Version 6.0, contains Table 1-8 describing the CRB data structure fields. The description in the table for the CRB$L_INTD field is confusing and needs clarification. Replace the first two sentences in the CRB$L_INTD description as follows:

Field Name	Description
CRB$L_INTD	Portion of the interrupt transfer vector block that stores executable code, driver entry points, and I/O adapter information. This 10-longword area is overlaid with the contents of the interrupt transfer vector block that starts at VEC$L_INTD (offset 16) as described in Section 1.7.1. It contains pointers to the driver's . . .

4.12.1.2.3 SCDRP Data Structure SCSI Flags

V6.1

Chapter 1 of the *OpenVMS VAX Device Support Reference Manual*, Version 6.0, contains a section describing the SCDRP data structure SCSI flags.

In the SCDRP$L_SCSI_FLAGS field description for bit SCDRP$V_LOCK, make the following correction:

Change: SCDRP$VLOCK

To: SCDRP$V_LOCK

4.12.1.2.4 SPI$CONNECT Macro

V6.1

Chapter 2 of the *OpenVMS VAX Device Support Reference Manual*, Version 6.0, contains a section describing the SPI$CONNECT macro.

In the table listing the required inputs, add:

R4 Address of the SPDT

In the values returned in R3, the SPDT$M_CMDQ bit was added to the port capability mask (SPDT$L_PORT_FLAGS). When set, SPDT$M_CMDQ indicates that the port driver supports command queuing I/O.

In the return values table listing R3 and the mask bits (after SPDT$M_LUNS), add:

SPDT$M_CMDQ Supports command queuing I/O

4.12.1.2.5 SPI$GET_CONNECTION_CHAR and SPI$SET_CONNECTION_CHAR Macros

V6.1

Chapter 2 of the *OpenVMS VAX Device Support Reference Manual*, Version 6.0, contains sections describing the SPI$GET_CONNECTION_CHAR and SPI$SET_CONNECTION_CHAR macros. Appended to the macro characteristics buffer is longword #12 for SCSI-2 support.

At the end of the characteristics buffer table in these macro descriptions, add the longword #12 information as follows:

12 SCSI-2 device characteristic status bits. Bits of this longword are defined as follows:

Bit	Description
0	When set, (SCDT$V_SCSI_2) indicates the device connection is SCSI-2 conformant.
1	When set, (SCDT$V_CMDQ) indicates the device connection supports command queuing.

4.12.1.2.6 $EQULST Macro

V6.1

Chapter 2 of the *OpenVMS VAX Device Support Reference Manual*, Version 6.0, contains a section describing the $EQULST macro.

In the parameter description for **symbol,value** insert the phrase *in decimal* as follows:

" . . . and value specifies in decimal the value of the symbol." ♦

4.13 I/O User

The following sections contain release notes pertinent to the I/O user.

4.13.1 Problems and Restrictions

This section describes known problems with I/O write operations.

4.13.1.1 I/O Write Operations Interrupted by System Failure

V6.0

The OpenVMS operating system ensures that when an I/O write operation returns a successful completion status, the data is available on the disk or tape media.

OpenVMS guarantees atomicity for single-block I/O write operations to DIGITAL Storage Architecture (DSA) drives.[1] However, if a system failure occurs while a multiple block I/O write operation is in progress, the OpenVMS operating system does not guarantee the successful completion of the I/O write. When a failure interrupts a write operation, the data can be left in any one of the following conditions:

- The new data is written completely to the disk blocks on the media, but a completion status was not returned before the failure.
- The new data is partially written to the media so that some of the disk blocks involved in the I/O contain the data from the in-progress I/O write and other blocks contain the data that was present prior to the I/O write operation. Note that blocks updated with new data can be interspersed with blocks containing old data.
- The new data was never written to any of the disk blocks on the media.

Individual blocks can contain either all new data or all old data; a block is not partially updated when dealing with DSA devices.

To guarantee that an I/O write operation either completes successfully, or (in the event of failure) is redone or rolled back as if it were never started, applications must be supplemented by additional techniques to ensure data correctness and recovery. For example, using RMS Journaling for OpenVMS or database or file journaling and recovery techniques allows applications to automatically recover from failures such as:

- Permanent loss of the path between a CPU data buffer containing the data being written and the disk being written to during a multiple block I/O operation. Communication path loss can occur due to node failure or a failure of node-to-node communications. Note that transient failures are automatically recovered from during mount verification without requiring any further recovery techniques.

[1] Single block atomicity may not be guaranteed with SCSI devices. This includes the new HSZ and HSJ SCSI adapters that extend SCSI disk connectivity to the HSC family of devices. An OpenVMS operating system makes no distinction between the SCSI-adapted DSA devices and actual SCSI devices.

- Failure of a CPU (such as a system crash, halt, power failure) during a multiple block I/O write operation.
- Mistaken deletion of a file.
- OpenVMS RMS incomplete bucket write operation to an indexed file.
- Cancellation of an in-progress multiple block write operation.

Note that Volume Shadowing for OpenVMS promotes availability by providing access to data in the presence of media deterioration, communication path failure, or controller or device failure. However, volume shadowing should not be mistaken for a journaling product that tracks changes to files for data integrity.

4.13.2 Documentation Changes and Corrections

This section describes changes to I/O documentation.

4.13.2.1 *OpenVMS I/O User's Reference Manual*

V6.1

The following section should appear in the Terminal Driver chapter of the *OpenVMS I/O User's Reference Manual*.

Terminal Devices Supported

Table 4–6 lists the supported terminal devices for the Digital 2100 Server Model A500/600 MP and the DEC 2000 Model 300.

Table 4–6 Supported Terminal Devices

Terminal Interface	Number of Lines	Output		Split Speed	Bus	International Modem Control
		Silo	DMA			
Digital 2100 Server Model A500/600MP	2	No	No	No	None	Full
Digital 2100 Server Model A500/600MP[1]	4, 8	Yes	No	No	EISA	Full
DEC 2000 Model 300	2	No	No	No	None	Full
DEC 2000 Model 300[1]	4, 8	Yes	No	No	EISA	Full

[1] Optional multiplex serial DIGIboard PC/X™ adapter card. You can daisy chain up to 4 boards in one system, resulting in 16, 32, or 64 ports.

4.14 Librarian Utility (LIBRARIAN)

The following sections contain release notes pertaining to the Librarian utility (LIBRARIAN).

4.14.1 Problems and Restrictions

This section describes known LIBRARIAN problems and restrictions.

4.14.1.1 PGFLQUOTA Should Exceed 23000 (Alpha Only)

V1.5

Alpha

The OpenVMS Alpha LIBRARIAN sometimes does not inform you of errors during compression, data reduction, or data expansion operations. This problem occurs if the account or process in which the LIBRARIAN is running has a low PGFLQUOTA process quota. Operation failure is not readily apparent because the $PUTMSG system service always returns a status of SS$_NORMAL, even when the system service fails. However, when a failure occurs, the LIBRARIAN returns a status other than success.

To work around this problem, run the compression, data reduction, or data expansion operation in an account with a PGFLQUOTA process quota greater than 23000. In addition, ensure that your command procedures check the return status from the LIBRARY command. ♦

4.15 Linker Utility

The following sections contain release notes pertaining to the OpenVMS Alpha Linker utility (linker).

4.15.1 Problems and Restrictions

This section describes known linker problems and restrictions.

4.15.1.1 Fixup Information Not Generated (Alpha Only)

V1.0

Alpha

The linker has been modified so that a new error message informs you at link time that global symbols from shareable images are being placed into byte- or word-sized fields by the linker. (Word- and byte-sized stores of global symbols do not generate fixup information. Fixup information is required when linking against shareable images.) When this situation occurs, an error message is printed, and image production is inhibited.

The following example shows this new error message:

```
%LINK-E-NOFIXSYM, unable to perform WORD fixup for symbol TPU$_OPTIONS
        in psect $PLIT$ in module TEST_MODULE file USER:[ACCOUNT]TEST.OLB;1
```

To work around this restriction, move the symbolic value into the desired location at run time rather than at link time. For example, in MACRO, rather than performing .WORD TPU$_OPTIONS, use the instruction MOVW #TPU$_ OPTIONS,*dest*. ♦

4.15.1.2 Shareable Image Psects Pointed to by Symbol Definitions (Alpha Only)

V1.0

Alpha

The linker cannot overlay program sections that are referenced by symbol definitions with shareable image program sections of the same name. Symbol definition records that contain the index of an overlaid program section are generated by the C compiler when the relaxed ref-def extern model is used (the default). Shareable image program sections are created when you link a shareable image and use the PSECT keyword in your SYMBOL_VECTOR option.

If the linker detects this condition, it issues the following error:

```
%LINK-E-SHRSYMFND, shareable image psect <name> was pointed to by a symbol definition
%LINK-E-NOIMGFIL, image file not created
```

The link continues, but no image is created. To work around this restriction, change the symbol vector keyword to DATA, or recompile your C program with the qualifier /EXTERN=COMMON. ♦

4.15.2 Corrections

This section describes corrections to the linker.

4.15.2.1 /DEBUG Qualifier

V6.2

Prior to the last release, the linker was modified to avoid multiple relocations and fixups to the same location from occurring during the processing of overlaid program sections. A problem has since been reported about "lost" relocations and fixups to shareable images when linked using the /DEBUG qualifier.

The problem was caused by the linker erroneously canceling deferred relocations or fixups to the image while processing debug records to generate the Debug Symbol Table (DST). Since the debug context is a separate address space from the image proper, a write to a virtual address from the debug space was removing relocations or fixups from the image located at the same virtual address.

This problem has been corrected.

4.15.3 Documentation Changes and Corrections

This section describes corrections to linker documentation.

4.15.3.1 *OpenVMS Linker Utility Manual*

V1.5

The *OpenVMS Linker Utility Manual* incorrectly describes /NONATIVE_ONLY to be the default behavior of the linker.

The /NATIVE_ONLY qualifier to the LINK command directs the linker *not* to pass along the procedure signature block (PSB) information, created by the compilers, in the image it is creating. This is the default behavior of the linker. For more information, refer to Section 4.20.1.1.

4.16 Lock Manager

The following sections contain release notes pertaining to the lock manager.

4.16.1 Changes and Enhancements

This section describes changes or enhancements to the software.

4.16.1.1 LCKMGR Spin Lock (Alpha Only)

V6.1

Alpha

The synchronization for the OpenVMS lock manager has changed with this release of OpenVMS Alpha. A new spin lock with a name of LCKMGR is now used to synchronize the OpenVMS lock manager for standalone OpenVMS Alpha machines. Alpha systems running as part of a VMScluster still synchronize with the SCS (IOLOCK8) spin lock. Use of the LCKMGR spin lock in a VMScluster system will correctly lock the SCS spin lock.

This change has no impact for users of the system service interfaces of $ENQ[W], $DEQ, and $GETLKI[W].

If you have code that currently accesses OpenVMS internal lock manager data structures such as LKBs or RSBs, you will need to modify your software to correctly synchronize with the lock manager by using the LCKMGR spin lock. This change does not affect OpenVMS VAX software. ♦

4.17 LTDRIVER

The following sections contain release notes pertaining to the LTDRIVER.

4.17.1 Problems and Restrictions

This section describes known LTDRIVER problems and restrictions.

4.17.1.1 CANCEL SELECTIVE Cannot Cancel IO$_TTY_PORT Functions

V6.1

In prior releases, LTDRIVER did not set the "extended DDT" bit; therefore, the POSIX function CANCEL SELECTIVE did not work with LTDRIVER. This has been fixed, but a restriction remains.

Although this fix allows $QIO reads and writes to be selectively canceled, any $QIO done to the port driver (that is, with the IO$_TTY_PORT function modifier—like a LAT connect $QIO) *cannot* be canceled with CANCEL SELECTIVE. This problem will be addressed in a future OpenVMS release.

4.18 MACRO-32 Compiler for OpenVMS Alpha (Alpha Only)

Alpha

The following sections contain information pertaining to the MACRO-32 Compiler.

4.18.1 Problems and Restrictions

This section describes problems and restrictions pertaining to the MACRO-32 Compiler for OpenVMS Alpha.

4.18.1.1 .ASCID Directive

V6.2

The length of the argument to a .ASCID directive is limited to 996 characters when using the OpenVMS Alpha MACRO-32 Compiler. No such restriction exists in the VAX MACRO Assembler.

This newly discovered difference will remain a restriction of the compiler.

4.18.1.2 .GLOBAL_LABEL Directive

V6.2

Labels declared with the .GLOBAL_LABEL directive can be used as the **newpc** argument in calls to the $UNWIND (Unwind Call Stack) System Service because it allows the address of the label to be stored.

However, there is no provision in the compiler to automatically adjust the stack pointer at such labels to remove arguments passed on the stack or compensate for stack alignment. If the call stack is unwound back to an alternate PC in the calling routine, the stack may still contain arguments and alignment bytes, and any stack-based references that expect this adjustment to the caller's original stack depth (which happened automatically on VAX) will be incorrect.

Code that contains labels declared with this directive that are to be used as alternate PC targets for $UNWIND must be examined carefully to ensure correct behavior, with particular emphasis on any references based on the stack pointer.

4.18.1.3 Floating-Point Return Values in R0

V6.1

A MACRO program that calls out to a routine and expects a floating-point return value in R0 may require a "jacket" between the call and the called routines to move the returned value from floating-point register 0 to R0.

4.18.1.4 INSV Instructions Negate Granularity Preservation

V6.1 INSV instructions do not generate code that correctly preserves granularity when granularity preservation is turned on.

4.18.1.5 .JSB_ENTRY Directive

V6.1 For procedures declared with the .JSB_ENTRY directive, the MACRO-32 Compiler automatically generates a null frame procedure descriptor, independent of debug or optimization qualifiers. The null frame procedure descriptor allows for debugging of problems with the linkage itself.

Because no new context is set up by a null frame procedure, a side effect is that there is no guarantee of completely accurate debugger information about such procedures in response to SHOW CALLS and SHOW STACK commands. For example, the line number in the called null procedure (to which a JSB is done) may be reported as the line number in the calling procedure from which the JSB is issued.

4.18.1.6 Floating-Point and Packed-Decimal Instruction Arguments

V1.5 Because packed-decimal instructions and floating-point instructions are implemented by means of macros, there is one restriction on the format of the arguments. In a macro invocation, an initial circumflex (^) is interpreted to mean that the parameter is a string and the character immediately following the circumflex is the string delimiter. Because of this, you cannot use arguments that begin with an operand type specification, such as ^x20(SP). Note that immediate mode arguments, such as #^XFF, can use an operand type specification because the circumflex is not the initial character.

4.18.1.7 Multiple Object Files Compilation

V1.5 The MACRO-32 Compiler does not support the creation of separate object files from multiple source files specified in one command. It does support the creation of one object file from multiple source files specified in one command.

4.18.2 Documentation Changes and Corrections

This section describes changes in the MACRO–32 compiler's documentation.

4.18.2.1 *Migrating to an OpenVMS AXP System: Porting VAX MACRO Code*

The following sections contain information that was included in the OpenVMS Alpha Version 6.1 Bookreader update of *Migrating to an OpenVMS AXP System: Porting VAX MACRO Code*, but which has not yet been published in the printed version of the manual.

4.18.2.1.1 Entry Point Declarations

V6.1 Any code label that is a possible target of a CALLS, CALLG, JSB, BSBW, or BSBB instruction must be declared as an entry point. In addition, any code label must be declared as an entry point using a .JSB_ENTRY or .JSB32_ENTRY directive if:

- The label can be the target of a global (cross-module) JMP, BRB, or BRW instruction
- The label can be the target of an indeterminate branch (such as BRB @(R10)), where the address of the label has been stored in R10, even if the reference and the label are within the same module
- The address of the label is stored in a register or memory location, even if it is never accessed by the current module

The OpenVMS calling standard for Alpha computers does not provide a way to access indeterminate code addresses directly. All such accesses are accomplished using a procedure descriptor to describe the routine and the code address. When a code label address is stored, the compiler does not know if that address will be referenced only by the current module, or whether it may be accessed by another MACRO module or another module written in another language. Whenever the MACRO–32 compiler stores a code address, it always stores the procedure descriptor for that address so that other code can access it correctly. For a procedure descriptor address to exist, the label must be declared as an entry point.

Likewise, when a stored address is used as a destination, the compiler does not know where that address came from, so it always assumes that the stored address is the address of a procedure descriptor and uses that descriptor to pass control to the routine.

4.18.2.1.2 BUG*x* Instruction Is Not Supported

V1.5 The BUG*x* instruction is not supported by the compiler. This instruction was inadvertently omitted from the list of unsupported instructions in *Migrating to an OpenVMS AXP System: Porting VAX MACRO Code*.

4.18.2.1.3 /FLAGGING=INSTRUCTIONS Qualifier

V1.5 When /FLAGGING=INSTRUCTIONS is enabled, absolute addresses detected by the compiler are flagged. For example, MOVL R0, 200 compiles correctly (updating memory location 200), but the desired absolute address may be different on an Alpha computer. As a result, the informational message CHKABSADR is reported. This does not also apply to the instruction M*x*PR, as stated in the description of the /FLAGGING=INSTRUCTIONS qualifier in Appendix A.

If you do not want these information messages reported, specify the /NOFLAGGING=INSTRUCTIONS qualifier.

4.18.2.1.4 TRACEBACK and DEBUG Options Restriction Removed

V1.5 The restriction that pertained to the qualifiers /DEBUG, /DISABLE, and /ENABLE and to the directives .DISABLE and .ENABLE, which is documented in Appendixes A and B, has been removed.

Contrary to the documentation, you can initially enable the TRACEBACK and DEBUG options with the /ENABLE qualifier. You can then turn the options off and on with the .DISABLE and .ENABLE directives for whichever code sections you want. However, if the /DEBUG qualifier is used in the command line, it overrides /ENABLE=(DEBUG,TRACEBACK) and /DISABLE=(DEBUG,TRACEBACK), regardless of their position on the command line. ♦

4.19 Mathematics (MTH$) Run-Time Library

The following sections contain release notes pertaining to the Run-Time Mathematics Library (MTH$).

4.19.1 Problems and Restrictions

This section describes known MTH$ problems and restrictions.

4.19.1.1 Linking Images to Run on Previous OpenVMS VAX Versions (VAX Only)

V6.1

VAX

This version of OpenVMS VAX provides updated versions of the Mathematics Run-Time Library (RTL) images MTHRTL.EXE, UVMTHRTL.EXE, and VMTHRTL.EXE that contain new entry points in support of DEC Fortran Version 6.0. (UVMTHRTL.EXE is an alternate form of MTHRTL.EXE; references to MTHRTL.EXE in the following paragraphs also apply to UVMTHRTL.EXE.)

Due to the large number of entry points added to MTHRTL.EXE, that image's transfer vector was extended and its global section match identifier incremented. This means that images linked against the new version of MTHRTL.EXE will not run on a system running a previous version of OpenVMS VAX, unless that system has also installed DEC Fortran Version 6.0. In addition, images linked against the new MTHRTL.EXE cannot be translated to run on OpenVMS Alpha using DECmigrate.

To link an image so that it will run on a previous version of OpenVMS VAX, create a directory that contains saved copies of the .EXE and .OLB files from the SYS$LIBRARY directory of the earliest version you wish to support, and define the logical name SYS$LIBRARY to point to that directory before linking. Because OpenVMS VAX also defines a system logical name MTHRTL to refer to either MTHRTL.EXE or UVMTHRTL.EXE, you must also define MTHRTL as a logical name in the process or job table to point to the copy in the directory of older images. For example:

```
$ DEFINE/USER SYS$LIBRARY disk:[OLD_SYSLIB]
$ DEFINE/USER MTHRTL SYS$LIBRARY:MTHRTL.EXE
$ LINK ...
```

Images to be translated using DECmigrate should be linked against the SYS$LIBRARY files of OpenVMS VAX Version 5.5-2 or earlier. ♦

4.19.1.2 Compatibility Between VAX and Alpha Mathematics Libraries (Alpha Only)

V1.0

Alpha

Mathematical applications using the standard OpenVMS call interface to the OpenVMS Run-Time Mathematics (MTH$) Library need not change their calls to MTH$ routines when migrating to an Alpha system. Jacket routines map MTH$ routines to their math$ counterparts in the Digital Portable Mathematics Library (DPML) for OpenVMS Alpha. However, there is no support in the DPML for calls made to JSB entry points and vector routines. Note that DPML routines are different from those in the OpenVMS Run-Time Mathematics (MTH$) Library. You should expect to see small differences in the precision of the mathematical results.

If one of your goals is to maintain compatibility with future libraries and to create portable mathematical applications, Digital recommends that you use the DPML routines available through the high-level language of your choice (for example, DEC Fortran and DEC C) rather than using the call interface. Significantly higher performance and accuracy are also available to you with DPML routines.

See the *Digital Portable Mathematics Library* manual for more information about DPML. ♦

4.20 Migration to OpenVMS Alpha Systems (Alpha Only)

Alpha

The following sections contain release notes pertaining to migration from OpenVMS VAX systems to OpenVMS Alpha systems.

4.20.1 Documentation Changes and Corrections

This section describes corrections to migration documentation. Also see Section 4.18.2 for corrections to *Migrating to an OpenVMS AXP System: Porting VAX MACRO Code*.

4.20.1.1 *Migrating to an OpenVMS AXP System: Recompiling and Relinking Applications*

V1.5

Migrating to an OpenVMS AXP System: Recompiling and Relinking Applications incorrectly describes /NONATIVE_ONLY to be the default behavior of the linker.

The /NATIVE_ONLY qualifier to the LINK command directs the linker *not* to pass along the procedure signature block (PSB) information, created by the compilers, in the image it is creating. This is the default behavior of the linker.

PSB information is necessary if the image you are creating in the link operation calls translated VAX images (but not if translated VAX images call it). To include PSB information in the image, you must compile the original program sources with the appropriate compiler switch (as indicated by the compiler documentation) and specify the /NONATIVE qualifier when linking the program's object modules. The image activator uses the PSB information in the image to create jacket routines, which allow native Alpha images to work with translated VAX images. ♦

4.21 POLYCENTER Software Installation Utility

The notes in this section pertain to the POLYCENTER Software Installation utility. Also see Section 3.22 for notes about this utility that are of interest to system managers.

4.21.1 Problems and Restrictions

This section describes known problems and restrictions with using the POLYCENTER Software Installation utility to create software kits. Problems and restrictions of interest to system managers are described in Section 3.22.1.

4.21.1.1 Account Removal Problem

V6.1

Removing a product with the POLYCENTER Software Installation utility results in the removal of accounts specified by *account* statements. The utility should not do this in all cases because the file SYSUAF.DAT may be clusterwide or local to a single system disk or groups of system disks.

If you install software on a particular system disk, any *account* statements create the necessary accounts. If you install the software on a second system disk that shares the same SYSUAF.DAT file, the account already exists and does not need to be created.

When you remove a product, the utility always removes the account, regardless of whether the SYSUAF.DAT file is shared by another system disk.

The same problem exists with the *rights identifier* statement and the file RIGHTSLIST.DAT.

Digital expects to correct these problems in a future release.

4.21.1.2 Alphanumeric Options Not Supported

V6.1

You cannot allow installers to specify alphanumeric configuration choices using the POLYCENTER Software Installation utility. The *option* statement allows only Boolean configuration choices.

Digital expects to correct this problem in a future release.

4.21.1.3 Alternate File Placement Not Supported

V6.1

You cannot specify an alternate location for some of your product files using the POLYCENTER Software Installation utility.

Digital expects to correct this problem in a future release.

4.21.1.4 Command Procedure Behavior

V6.1

The following product description file (PDF) statements use command procedures to perform the creation and deletion of managed objects:

- *account*
- *network object*
- *rights identifier*

In a future version, the POLYCENTER Software Installation utility may create and delete these managed objects directly without the use of command procedures. If this is the case, these statements will continue to function, but the command procedures may not be maintained or shipped with future versions of the utility.

4.21.1.5 Conflict Error Reporting

V6.1

Some error messages indicating managed object conflicts do not identify the related products.

4.21.1.6 Error Reporting

V6.1

When packaging a kit, you may receive many different errors (for example, wrong keywords or missing directives) in your product description file (PDF). In several cases, the utility error messages do not contain enough information to determine the problem with the PDF.

Digital expects to correct this problem in a future release.

4.21.1.7 File Generation Restrictions

V6.1

The generation option to the *file* statement does not work correctly under some circumstances.

For example:

```
-- PDF #1:
product DEC VAXVMS TEST1 V1.0 full ;
file [SYSMGR]TEST.EXE generation 1 ;
end product ;

-- PDF #2:
product DEC VAXVMS TEST2 V1.0 full ;
file [SYSMGR]TEST.EXE generation 2 ;
end product ;
```

Installing TEST1 then TEST2 works correctly. However, if you remove TEST2, generation one of the TEST.EXE file is not reinstalled and the utility displays a message about replacement material being unavailable.

One workaround is to have TEST2 use the *execute install* statement to execute a procedure that saves previous versions of TEST.EXE. A corresponding *execute remove* could restore it.

Another related problem is that you can remove TEST1 and then TEST2, but only in that order. The previous example shows removing TEST2 first which is the logical order (since it is the reverse order of installation).

Note that if you install TEST2 then TEST1, the installation of TEST1 generates an error message about replacement material being unavailable. The utility should give a better error message that indicates that a newer generation of the file is already installed on the system.

Digital expects to correct these problems in a future release.

4.21.1.8 File Statement Generation Limit

V6.1 The maximum value allowed for the generation option on the file statement is 4294967295 (2**32-1). If you specify a larger number, the POLYCENTER Software Installation utility incorrectly uses the overflow value without warning.

Digital expects to correct this problem in a future release.

4.21.1.9 Information Statement Problem

V6.1 Product Description File (PDF) *information* statements that you want to display to the user during the configuration phase are not displayed if the user enters Yes (the default answer) to the following question:

```
Do you want all the default values for this product? [YES]
```

Digital expects to correct this problem in a future release.

4.21.1.10 Multiple Execute Remove Statements

V6.1 There is a problem with the *execute remove* statement where only the first one executes during a remove operation. However, all of the *execute install* statements execute during an install operation.

Digital expects to correct this problem in a future release.

4.21.1.11 Multiple Operating Scopes Not Supported

V6.1 Multiple operating scopes are not supported and may not work correctly in certain circumstances.

4.21.1.12 Network Object Statement Problem

V6.1 The command procedure that creates network objects for the *network object* statement asks the product developer for the network object number. This is information that the installer, not the product developer, has access to. Also, the utility does not provide a way for the installer to enter this information.

4.21.1.13 Package Operation Constraints

V6.1 A typical task when creating software kits is to create a sequential kit by performing a package operation. If you use the same directory for the source and destination for your package operation, you can create problems for future install operations that use the directory as a source.

If both a sequential kit (with file type .PCSI) and a product description file for a reference kit (with file type .PCSI$DESCRIPTION) are in the same directory, an install operation that points to that source area always opens the file with file type .PCSI$DESCRIPTION. This can cause errors locating product material if

it is not in the same place. There is no way to specify the sequential kit for the installation when a reference kit for the same product is present.

To avoid this problem, use separate directories for the source and destination of your package operations.

4.21.1.14 Partial Kit Restrictions

V6.1

Kits of type partial do work correctly under many circumstances. Digital recommends that you avoid creating partial kits with this version of the POLYCENTER Software Installation utility because of the following problems and restrictions:

- The utility does not require that a full kit of the product has already been installed.
- If a full product kit is not found, the utility attempts to attach the partial kit to the operating system as its parent.
- When removing a partial kit that has the operating system as its parent, the utility also removes the operating system without displaying any messages.

If possible, use patch or mandatory update kit types rather than partial kits.

Digital expects to correct these problems in a future release.

4.21.1.15 Rights Identifier Problem

V6.1

Refer to Section 4.21.1.1 for a description of this problem.

4.21.1.16 "Uses" Clause File Specification Restriction

V6.1

You cannot specify the same file in the "uses" clause of two separate execute statements in the same product description file (PDF), for example:

```
product DEC AXPVMS MYSTUFF full;
     ...
     execute install "@doit thisway" uses [000000]doit.com;
     execute install "@doit thatway" uses [000000]doit.com;
     ...
end product;
```

The file is deleted before the second use.

Digital expects to correct this problem in a future release.

4.21.2 Documentation Changes and Corrections

This section describes changes in the utility's documentation.

4.21.2.1 DCL Help: PRODUCT PACKAGE Command

V6.2

The *POLYCENTER Software Installation Utility Developer's Guide* has not been updated for the OpenVMS Version 6.2 release, but the description of the PRODUCT PACKAGE command and its qualifiers has been significantly enhanced in online Help. Enter the HELP PRODUCT PACKAGE command at the DCL prompt to view this updated documentation.

4.21.2.2 *POLYCENTER Software Installation Utility User's Guide*

V6.2

Information formerly included in the *POLYCENTER Software Installation Utility User's Guide* has been revised and incorporated into the *OpenVMS System Manager's Manual: Essentials*.

4.21.2.3 *POLYCENTER Software Installation Utility Developer's Guide*

The following sections describe corrections to the *POLYCENTER Software Installation Utility Developer's Guide*.

4.21.2.3.1 File Statement

V6.2

In the *POLYCENTER Software Installation Utility Developer's Guide*, the **file** statement is described in the Product Description Language (PDL) reference section. Replace the description of the **[no]generation** option as follows:

[no]generation *generation*
Specifies the generation number of the file as an unsigned integer from 0 to 4294967295. Using the **[no]generation** clause is equivalent to specifying a value of 0; both mean the file has no explicit generation number. The default for the file statement is **[no]generation**. Refer to the Description section for the meaning of this value.

4.21.2.3.2 Scope Statement

V6.1

In the *POLYCENTER Software Installation Utility Developer's Guide*, the **scope** statement is described in the Product Description Language (PDL) reference section and in Appendix B. The following notes apply to the scope statement:

- *Bootstrap Scope* should be used for products that use device drivers, especially those drivers that must be read by the primitive file system. Because files in Bootstrap Scope are read by the primitive file system, they are read when not synchronized with the file system on other cluster members that might access the same disk. Therefore, those files must retain stable positions as long as the disk is in use from *any* system and must not be manipulated by online disk defragmentation operations, even those that use the MOVEFILE primitive.
- *Product Scope* is the default scope for most objects; therefore, it is generally unnecessary to specify Product Scope.
- *Operating Scope* is not implemented for this version of the POLYCENTER Software Installation utility.

4.22 Run-Time Libraries

The following sections contain release notes pertaining to run-time libraries.

4.22.1 Problems and Restrictions

This section describes known run-time library problems and restrictions.

4.22.1.1 Run-Time Libraries Not Included in OpenVMS Alpha Systems (Alpha Only)

V1.5

Alpha

The run-time libraries listed in Table 4–7 are not included in this version of OpenVMS Alpha.

Table 4–7 Run-Time Libraries Not Included on OpenVMS Alpha Systems

DBGSSISHR	DEBUG item, replaced by SYS$SSISHR on OpenVMS Alpha
DNS$RTL	No DNS in OpenVMS Alpha
DNS$SHARE	No DNS in OpenVMS Alpha
VBLAS1RTL	No support for VAX vector programs
VMTHRTL	No support for VAX vector programs

Most run-time libraries that were available in OpenVMS VAX Version 5.5-2 are available in this version of OpenVMS Alpha. The OpenVMS VAX Version 5.5-2 libraries that are not available are either not being ported to OpenVMS Alpha or are planned for a later release of OpenVMS Alpha.

For example, the vector math libraries VBLAS1RTL and VMTHRTL are not available in OpenVMS Alpha because there is no support on OpenVMS Alpha for programs that use the VAX vector instructions. ♦

4.23 Screen Management (SMG$) Facility

The following sections contain release notes pertaining to the Screen Management (SMG$) Facility.

4.23.1 Changes and Enhancements

This section describes changes and enhancements to the software.

4.23.1.1 VT500 Series Terminal Support

V6.2

The Screen Management (SMG$) Facility implements support for VT500 series terminals through the use of a third group of characteristics. The following specific changes were made to SMG$ to support VT500 series terminals.

New Boolean Capabilities

Two new Boolean capabilities have been added for use as TERMTABLE entries:

OpenVMS Name	Used by SMG	Description
ANSI_COLOR	N	If set, terminal conforms to ANSI color programming standards
DEC_CRT_5	N	If set, terminal conforms to DIGITAL VT500 family standards

Changes to SMG$ Routines

The following SMG$ routines have been changed as described below to support VT500 series terminals:

SMG$GET_KEYBOARD_ATTRIBUTES The following new symbolic name provides access to VT500 series keyboard information table values through the **keyboard-info-table** argument.

SMG$L_DEV_DEPEND3 Specific characteristics 3 (longword).

SMG$GET_PASTEBOARD_ATTRIBUTES The following new symbolic name provides access to VT500 series pasteboard attribute values through the **pasteboard-info-table** argument.

SMG$L_DEVDEPEND3	Specific characteristics 3 (longword).

SMG$SET_TERM_CHARACTERISTICS Three new optional arguments change terminal characteristics and retrieve the current terminal characteristics for VT500 series terminals. The three arguments are **on-characteristics3**, **off-characteristics3**, and **old-characteristics3**.

The new syntax is:

SMG$SET_TERM_CHARACTERISTICS pasteboard-id
[,on-characteristics1] [,on-characteristics2]
[,off-characteristics1] [,off-characteristics2]
[,old-characteristics1] [,old-characteristics2]
[,on-characteristics3] [,off-characteristics3]
[,old-characteristics3]

The three new arguments are defined as follows:

on-characteristics3
mask_longword
longword (unsigned)
read only
by reference
Bit mask that specifies the terminal characteristics to be set from $TT3DEF. The **on-characteristics3** argument is the address of an unsigned longword that contains the bit mask.

off-characteristics3
mask_longword
longword (unsigned)
read only
by reference
Bit mask that specifies the terminal characteristics to be reset from $TT3DEF. The **off-characteristics3** argument is the address of an unsigned longword that contains the bit mask.

old-characteristics3
mask_longword
longword (unsigned)
read only
by reference
Retrieves the current terminal characteristics in the third group. The **old-characteristics3** argument is the address of an unsigned longword that contains the bit mask.

4.24 Security

The following sections contain release notes pertaining to security.

4.24.1 Changes and Enhancements

This section describes changes or enhancements to the software.

4.24.1.1 SYS$EXAMPLES:HASH_PASSWORD.MAR

V6.1

Two changes were made to the site-specific password hash algorithm example code in SYS$EXAMPLES:HASH_PASSWORD.MAR. Sites using this file as a basis for a site-specific password hash algorithm must recompile and relink their SYS$HASH_PASSWORD executive loadable images based on the new template.

If you modified the Digital-supplied preamble code in the example, note the functional changes listed below. As before, site-specific algorithm dispatching generally replaces a "NOP" instruction in the template.

- The value of UAF$C_PREFERRED_ALGORITHM has been changed from 3 to 127. The new value will remain constant from now on. Existing binary images that reference the preferred algorithm using the old value will continue to function, but will need to be recompiled and relinked if the preferred algorithm were ever to change. With this change, such images will have to recompile only once.
- User-mode probing of the formal parameters is now conditionally assembled based on the PROTECT conditional assembly prefix. If the service is being assembled as an exec-mode service, PROBEs are still used. Otherwise, the system condition handler EXE$SIGTORET is used to turn user-mode signals into their corresponding return status values. This change corrects a problem seen when using $HASH_PASSWORD with static watchpoints in the OpenVMS Debugger.

4.25 STARLET Data Structures and Definitions for C Programmers (Alpha Only)

V1.0

Alpha

OpenVMS Alpha Version 1.0 includes a new file, SYS$STARLET_C.TLB, that contains all the .H files that provide STARLET functionality equivalent to STARLETSD.TLB. The file SYS$STARLET_C.TLB, together with DECC$RTLDEF.TLB now shipping with the DEC C Compiler, replaces VAXCDEF.TLB that previously shipped with the VAX C Compiler. DECC$RTLDEF.TLB contains all the .H files that support the compiler and RTL, such as STDIO.H.

The following differences may require source changes:

- RMS structures

  Previously, the RMS structures FAB, NAM, RAB, XABALL, and so forth, were defined in the appropriate .H files as “struct RAB {...”, for example. The .H files to be supplied in OpenVMS Alpha Version 1.0 will define them as “struct rabdef {...”. To compensate for this difference, lines of the form “#define RAB rabdef” have been added. However, there is one situation where a source change is required because of this change. If you have a private structure that contains a pointer to one of these structures and your private structure is defined (but not used) before the RMS structure has been defined, you will receive compile-time errors similar to the following:

```
%CC-E-PASNOTMEM, In this statement, "rab$b_rac" is not a member of "rab".
```

This error can be avoided by reordering your source file so that the RMS structure is defined before the private structure. Typically, this involves moving around "#include" statements.

- LIB (privileged interface) structures

  Historically, three structures from LIB (NFBDEF.H, FATDEF.H, and FCHDEF.H) have been made available as .H files. These files were shipped as .H files in OpenVMS Alpha Version 1.0 and 1.5 (not in the new SYS$STARLET_C.TLB). In OpenVMS Alpha Version 6.1, the file SYS$LIB_C.TLB, containing all LIB structures and definitions, has been added. These three .H files are now part of that .TLB and are no longer shipped separately. Source changes may be required, as no attempt has been made to preserve any existing anomalies in these files. The structures and definitions from LIB are for privileged interfaces only and are therefore subject to change.

- Use of "variant_struct" and "variant_union"

  In the new .H files, "variant_struct" and "variant_union" are always used, whereas previously some structures used "struct" and "union". Therefore, the intermediate structure names cannot be specified when referencing fields within data structures.

  For example, the following statement:

  ```
  AlignFaultItem.PC[0] = DataPtr->afr$r_pc_data_overlay.afr$q_fault_pc[0];
  ```

  becomes:

  ```
  AlignFaultItem.PC[0] = DataPtr->afr$q_fault_pc[0];
  ```

- Member alignment

  Each of the .H files in SYS$STARLET_C.TLB saves and restores the state of "#pragma member_alignment".

- Conventions

  The .H files in SYS$STARLET_C.TLB adhere to some conventions that were only partly followed in VAXCDEF.TLB. All constants (#defines) have uppercase names. All identifiers (routines, structure members, and so forth) have lowercase names. Where there is a difference from VAXCDEF.TLB, the old symbol name is also included for compatibility, but users are encouraged to follow the new conventions.

- Use of Librarian utility to access the .H files

  During installation of OpenVMS Alpha Version 1.0, the contents of SYS$STARLET_C.TLB are not extracted into the separate .H files. The DEC C Compiler accesses these files from within SYS$STARLET_C.TLB, regardless of the format of the #include statement. If you want to inspect an individual .H file, you can use the Librarian utility, as in the following example:

  ```
  $ LIBRARY /EXTRACT=AFRDEF /OUTPUT=AFRDEF.H SYS$LIBRARY:SYS$STARLET_C.TLB
  ```

- Additional .H files included in SYS$STARLET_C.TLB

  In addition to the .H files derived from STARLET sources, SYS$STARLET_C.TLB includes .H files that provide support for DECthreads, such as CMA.H.♦

4.26 System Services

The following sections contain release notes pertaining to system services.

All system services are documented in the *OpenVMS System Services Reference Manual*.

4.26.1 Changes and Enhancements

The sections that follow describe changes and enhancements to the system services.

4.26.1.1 $INIT_VOL Has New INIT$_HOMEBLOCKS Item Code

V6.2

The INIT$_HOMEBLOCKS item code for the $INIT_VOL system service has been added to retain backward compatibility and to allow container file systems to make optimal use of disk space.

The INIT$_HOMEBLOCKS item code can have the following values:

- INIT$K_HOMEBLOCKS_GEOMETRY

  Causes the volume homeblocks to be placed on disk depending on the reported geometry of the disk. This is the behavior that occurred prior to OpenVMS Version 6.2.

- INIT$K_HOMEBLOCKS_FIXED

  Causes the volume homeblocks to be placed at fixed offsets on the disk regardless of geometry. This is the new default. It provides for disks that report different geometries, depending on the type of controller to which the disk is attached.

- INIT$K_HOMEBLOCKS_CONTIGUOUS

  Causes the volume homeblocks to be placed contiguously at the start of the disk. This allows container file systems to maximize the amount of contiguous space on the disk when used with the INIT$_INDEX_BEGINNING item code.

4.26.1.2 $LCKPAG and $LKWSET Limits Are Set by WSQUOTA

V6.2

The value set for WSQUOTA limits the number of pages that the system services $LCKPAG and $LKWSET can lock. The maximum value of WSQUOTA is 65536 (64K byte) pages.

4.26.1.3 $SNDJBC Has New SJC$_DISPLAY_USERNAME Item Code

V6.2

The SJC$_DISPLAY_USERNAME item code for the $SNDJBC system service has been added to make it easier for applications such as servers to submit print jobs for users. The item code has been available since OpenVMS VAX Version 5.5 and OpenVMS Alpha Version 1.5 but has not been documented. The item code is supported for these and later versions.

The SJC$_DISPLAY_USERNAME item code allows you to change the user name string without changing the requester's ARB (security access rights block). This item is useful for server applications that submit print jobs on behalf of users. It will modify the user name used in separator page output. It also specifies the user name string placed in the audit record when the symbiont does an access check on the input file. The access check will use the ARB of the requester, not the ARB of the user specified by the user name string. CMKRNL privilege is required.

4.26.1.4 $SYNCH and Wait Form Services

V6.2

For OpenVMS Version 6.2, the $SYNCH service has been modified to eliminate a problem that can cause a process to remain in event flag wait state after a service has completed. The problem is a race condition that can occur when the same event flag is concurrently used by multiple execution threads within the process. For this change to be effective, the caller must provide a status block (usually an IOSB). This change also applies to the wait form services, such as $QIOW, which call $SYNCH implicitly.

4.26.2 Problems and Restrictions

This section describes known problems and restrictions with system services.

4.26.2.1 $SNDJBC Batch Queue Limitation

VAX V6.0
Alpha V6.1

OpenVMS Version 6.0 allows multiple queue managers. If your system uses this feature to run batch queues on a separate queue manager from output queues, certain checks that would otherwise be performed for the SJC$_LOG_QUEUE item code of the $SNDJBC system service are not performed.

When batch and print queues are managed by the same queue manager, the queue manager checks to ensure that the queue specified with the SJC$_LOG_QUEUE is an output queue and that the user has access to the output queue. These checks are not made if the batch queue specified by the $SNDJBC service and the output queue specified by the SJC$_LOG_QUEUE item code are managed by different queue managers.

If you explicitly specify an output queue for the log file when submitting a batch job, be sure the queue you specify with the SJC$_LOG_QUEUE is an output queue and not a batch queue. Also, be sure that you have access to the printer queue.

4.26.2.2 $SUSPND Behaves Incorrectly in a Cluster Environment

VAX V6.0
Alpha V1.5

When the $SUSPND system service is called and the target process is on a different cluster node than that of the process calling the $SUSPND service, the kernel mode suspend flag (bit 0) is ignored. As a result, any suspend is treated as a supervisor-mode suspend.

Digital expects to fix this problem in a future release.

4.26.2.3 $FORMAT_AUDIT Width Argument Works Inconsistently

VAX V5.0
Alpha V1.0

The **width** argument to the $FORMAT_AUDIT system service does not work consistently. In most cases, if you specify both the **width** argument and the full format style (NSA$C_FORMAT_STYLE_FULL), $FORMAT_AUDIT ignores the **width** argument. The minimum width is 80 columns; lower values do not limit the width to less than 80. If you specify a width greater than 80 columns, most lines are not joined to use the full width.

In general, avoid using the **width** argument.

4.26.3 Documentation Changes and Corrections

The notes in this section describe changes and corrections to the *OpenVMS System Services Reference Manual*.

4.26.3.1 $CHKPRO System Service

V6.2 The following is a documentation correction to the $CHKPRO system service description; it replaces the CHP$_ACL item description:

CHP$_ACL

The buffer, specified by BUFADR and BUFLEN, contains one or more ACEs. The length of each ACE is contained in the first byte of the ACE. If the specified buffer length is greater than the sum of the ACE lengths, the byte following the last ACE must be zero (that is, the buffer can be terminated with a zero-length ACE).

The format of an ACE depends on the ACE type, which is the second byte of the ACE. The $FORMAT_ACL system service describes each ACE type and its format.

You can specify the CHP$_ACL item multiple times to point to multiple segments of an access control list, up to a maximum of 20 segments. Segments are processed in the order specified.

4.26.3.2 $TSTCLUEVT System Service

V6.2 Add the following information to the Description section for the $TSTCLUEVT system service in the *OpenVMS System Services Reference Manual: GETQUI–Z*:

> The $TSTCLUEVT system service allows an application to test itself and must be issued from within the same process as the application being tested. $TSTCLUEVT does not affect other processes in the cluster.

4.26.3.3 $CREATE_USER_PROFILE System Service

V6.1 The following corrections apply to the description of the $CREATE_USER_ PROFILE system service in the *OpenVMS System Services Reference Manual*:

- The **usrpro** argument is passed by descriptor, not by reference.
- The **usrprolen** argument has read/write access, not write-only access.
- The description of the CHP$_ADDRIGHTS item code is incorrect. It should read as follows:

  CHP$_ADDRIGHTS
  A rights list segment containing additional identifiers to be appended to the set of identifiers held by the user. A rights list segment is a list of quadword identifier/attributes pairs, each containing a longword identifier value, followed by a longword mask identifying the attributes of the holder. The **buflen** argument should be set to the total size, in bytes, of the rights list segment. The **bufadr** argument points to a descriptor that points to the first byte in the rights list segment (that is, the first byte of the first identifier value).

  This item code can be repeated to add up to 256 additional rights list segments. If more than 256 identifiers are granted to the user, $CREATE_ USER_PROFILE returns SS$_INSFMEM.

4.26.3.4 $GETQUI System Service

V6.1 The **context** argument to the $GETQUI system service is documented incorrectly in the *OpenVMS System Services Reference Manual*.

The documentation says the **context** argument has read-only access and is passed by value. In fact, the **context** argument has modify access and is passed by reference.

4.27 System Time Operation

The following sections contain release notes pertaining to the system time operation.

4.27.1 Documentation Changes and Corrections

This section includes new documentation information.

4.27.1.1 *OpenVMS Programming Concepts Manual*

V6.1

The following information should be included in the "System Time Operations" chapter of the *OpenVMS Programming Concepts Manual*.

The Date/Time Manipulation option provides date/time spelling support for four new languages. Users or application programmers can select the desired language by defining the logical name SYS$LANGUAGES. The new languages and their equivalent names are as follows:

Language	Equivalent Name
Chinese (simplified character)	Hanzi
Chinese (traditional character)	Hanyu
Korean	Hangul
Thai	Thai

Defining Date/Time Spelling

To define the spelling for Hanzi and Hanyu, define SYS$LANGUAGES as shown below, prior to invoking LIB$DT_STARTUP.COM:

```
$ DEFINE SYS$LANGUAGES HANZI, HANYU
$ @SYS$MANAGER:LIB$DT_STARTUP
```

Predefined Output Formats

Tables 4–8 and 4–9 list the new predefined date and time format logical names, their formats, and examples of the output generated using those formats.

Table 4–8 Predefined Output Date Formats

Logical Name	Format	Example
LIB$DATE_FORMAT_042	!Y4年!MNB月!DB日 !WAU	1994年3月7日 (一)
LIB$DATE_FORMAT_043	!Y4年!MNB月!DB日 !WU	1994年3月7日 星期一
LIB$DATE_FORMAT_044	!Y4年!MNB月!DB日 !WAU	1994年3月7日 (一)
LIB$DATE_FORMAT_045	!Y4年!MNB月!DB日 !WU	1994年3月7日 星期一
LIB$DATE_FORMAT_046	!Y4 년 !MNB 월 !DB 일 !WAU	1994 년 3 월 7 일 (월)
LIB$DATE_FORMAT_047	!Y4 년 !MNB 월 !DB 일 !WU	1994 년 3 월 7 일 월요일

ZK-7263A

Note

LIB$DATE_FORMAT_042 and LIB$DATE_FORMAT_043 support the DEC Hanzi coded character set.

LIB$DATE_FORMAT_044 and LIB$DATE_FORMAT_045 support the DEC Hanyu coded character set.

LIB$DATE_FORMAT_046 and LIB$DATE_FORMAT_047 support the DEC Hangul coded character set.

Table 4–9 Predefined Output Time Formats

Logical Name	Format	Example
LIB$TIME_FORMAT_021	!MIU!HB2时!MB分!SB秒	上午3时3分6秒
LIB$TIME_FORMAT_022	!MIU!HB2時!MB分!SB秒	上午3時3分6秒
LIB$TIME_FORMAT_023	!MIU !HB2 시 !MB 분 !SB 초	오전 3 시 3 분 6 초
		ZK-7262A

Note

LIB$TIME_FORMAT_021 supports the DEC Hanzi coded character set.

LIB$TIME_FORMAT_022 supports the DEC Hanyu coded character set.

LIB$TIME_FORMAT_023 supports the DEC Hangul coded character set.

Thus, to select a particular format for a date or time, or both, you can define the LIB$DT_FORMAT logical name using the following logicals:

- LIB$DATE_FORMAT_*nnn*, where *nnn* can range from 001 to 047
- LIB$TIME_FORMAT_*nnn*, where *nnn* can range from 001 to 023

4.28 Translated Image Environment (TIE) (Alpha Only)

V1.5

Alpha

Image translation is one means of migrating all or part of a VAX application to OpenVMS Alpha. The DECmigrate for OpenVMS AXP VAX Environment Software Translator utility (VEST) creates a translated image by converting a VAX executable or shareable image into a functionally equivalent Alpha image. VEST is a component of the optional layered product DECmigrate for OpenVMS AXP.

When a translated image runs on OpenVMS Alpha, the Translated Image Environment (TIE) provides the VAX environment required for the image to execute properly. The TIE consists of the shareable images TIE$SHARE and TIE$EMULAT_TV, which perform VAX complex instructions. For information on the role of image translation in a migration strategy, see the manuals *Migrating to an OpenVMS AXP System: Planning for Migration* and *DECmigrate for OpenVMS AXP Systems Translating Images*.

The following subsections discuss these topics:

- Interoperability between native and translated images
- Running translated images
- TIE statistics and feedback

Interoperability Between Native and Translated Images

V1.0

The TIE works together with other components of OpenVMS Alpha to enable native and translated images to interoperate, that is, to call one another. If you are developing applications or run-time libraries that rely on interoperability, you need to follow certain procedures when compiling, linking, or translating. See the first restriction described in Section 4.28.1.4. Table 4–10 provides pointers to documentation that describes the procedures.

Table 4–10 Interoperability Documentation

Goal	Reference
Ensuring interoperability between native and translated images	*Migrating to an OpenVMS AXP System: Recompiling and Relinking Applications*
	DECmigrate for OpenVMS AXP Systems Translating Images
Coordinating native and translated run-time libraries	*DECmigrate for OpenVMS AXP Systems Translating Images*

Running Translated Images

V1.0

Use the DCL RUN command to run a translated image. For example:

```
$ RUN FOO_TV.EXE
```

Note that the translated image does not run correctly unless OpenVMS Alpha includes the appropriate translated shareable images and run-time libraries. When you translate an image, VEST requires the image information files (IIFs—file type .IIF) corresponding to whichever images and libraries that the input image refers to. These .IIF files enable VEST to create a translated image that correctly refers to the translated versions of the shareable images and libraries. An image information file used at image translation must exactly correspond to the version of the translated shareable image or run-time library available on OpenVMS Alpha.

OpenVMS Alpha includes a set of translated run-time libraries and a matching set of image information files, which are listed in Section 4.29. Check these lists to determine if they include the libraries or shareable images referred to by images you want to translate and run. If OpenVMS Alpha does not include the required shared images or libraries, refer to the manual *DECmigrate for OpenVMS AXP Systems Translating Images*. This manual describes how to create and use image information files.

When a translated library has been replaced by a native version of the library, you need to define accordingly any logical names that point to it—that is, you need to redefine *image_TV* to *image*.

TIE Statistics and Feedback

V1.0 In addition to the TIE's run-time support function, TIE statistics and feedback can help to improve translated image performance:

- The TIE can display statistics about the run-time execution of translated images. These statistics describe the image's use of TIE resources and the interactions between images.
- The TIE can record information about VAX entry points discovered while interpreting VAX code. When you retranslate the image, VEST uses the information to find and translate more VAX code.

The manual *DECmigrate for OpenVMS AXP Systems Translating Images* describes these features in detail and explains how to define the logical names that enable and disable their use.

4.28.1 Problems and Restrictions

This section describes known problems and restrictions with the TIE.

4.28.1.1 Condition Handler Restriction

V1.5 There is a permanent restriction on the type of condition handler that can be established for both native and translated images. A native routine cannot establish a translated condition handler, nor can a translated routine establish a native condition handler. If a native or translated image violates this restriction, the run-time results are unpredictable.

4.28.1.2 Exception Handler Restrictions

V1.5 The following exception handler restrictions are permanent:

- Translated images with exception handlers that depend on receiving the correct program status longword (PSL) might not function properly. When exceptions are reported, the Alpha program status (PS) is reported in the signal array instead because there is no VAX PSL.
- Translated images with exception handlers that depend on modifying the PSL in the signal array do not function properly. The modified PSL is not propagated back to the faulting code.

4.28.1.3 Floating-Point Restrictions

V1.5 The following floating-point restrictions are permanent:

- In some cases, floating-point instructions operating on the same data generate a trap on an Alpha system but not on a VAX system. Specifically, VAX floating-point instructions on OpenVMS Alpha generate traps for the "dirty zeros" that VAX hardware can handle correctly. "Dirty zeros" are floating-point values that are alternate encodings for zero. To retain compatibility with translated code that performs operations using dirty zeros, the TIE includes a condition handler that corrects the dirty zeros and retries the floating-point operation. However, the handler succeeds only if the qualifier /PRESERVE=FLOAT_EXCEPTIONS was used when the image was translated.

  Images that were not translated with /PRESERVE=FLOAT_EXCEPTIONS and that perform an operation on a dirty zero incur an HPARITH exception with a summary status that has bit 1 set. If your translated application incurs one of these exceptions, retranslate with /PRESERVE=FLOAT_EXCEPTIONS. VAX dirty zeros commonly result from not initializing floating

data to 0. In this case, changes to source code may be necessary to port to OpenVMS Alpha an application that uses dirty zeros.

- Alpha D53 floating point (D_floating point as a 53-bit fraction instead of a 56-bit fraction) is VAX D_floating converted to G_floating representation. This conversion leads to the following problem. Consider the VAX instruction sequence:

```
MOVD    (SP),R2
MOVD    R2,-(SP)
```

VEST translates these VAX instructions into Alpha code like the following:

```
LDD     F2,0(R14)       ! Pickup D float
CVTDG   F2,F2           ! Convert to Canonical G Form with rounding
CVTGD   F2,F17          ! Convert back to D Form for storing
STD     F17,-8(R14)     ! Store the result
```

At run time, the VEST-generated code uses rounding to obtain the most accurate G_floating value when converting the D56 floating point to G canonical form. In some cases, the conversion to G canonical form may round up the D_floating value to create an exponent that cannot be represented in D_floating. When this happens, the CVTGD operation incurs an HPARITH trap with floating overflow as the summary reason.

If a translated image incurs this problem at run time, it needs to be retranslated with the VEST qualifier /FLOAT=D56_FLOAT to execute properly.

4.28.1.4 Interoperability Restrictions

V1.5 Note the following interoperability restrictions:

- A native routine that either calls or is called by a translated image must be compiled with the /TIE qualifier and be linked with the /NONATIVE_ONLY qualifier. Checking for interoperability between native and translated images occurs at run time. If the /TIE and /NONATIVE_ONLY qualifiers are not used to compile and link the native routine, an error occurs at run time when the the native routine and a translated image attempt to interoperate. If such an error occurs, recompile and relink the native routine appropriately.
- An access violation can occur at run time if a native routine that was not compiled with the /TIE qualifier makes an indirect call to a translated routine. The indirect call is made through a variable that contains the translated routine's address. When this happens, there is no autojacketing code in place to assist the native-to-translated call. The native code attempts to use the routine address as a native procedure descriptor. The code address of a native procedure is at offset PDSC$L_ENTRY, whose value is 8, from the base of the procedure descriptor. Because the translated routine address is treated as a procedure descriptor, the value at offset 8 from that address is used as the code to call. This usually results in an access violation.

  If you are encountering this problem, use a debugger to check the following:

  - Check that R27 points into a translated image.
  - Check that bits <31:2> of 8(R27) equal bits <31:2> of the access violation address. (All bits are not used because Alpha instructions are longword aligned.)
  - Check that R26 points into a native image.

- Check that -4(R26) is a JSR R26,(26) instruction.

If all these checks prove to be true, recompile the native routine with the /TIE qualifier to enable autojacketing at run time.

4.28.1.5 VAX C: Translated Program Restrictions

V1.5

The following translated VAX C program restrictions are permanent:

- If a program uses the VAX C RTL routine brk( ) to release dynamic memory (that is, a break address lower than the current break address is requested), the next attempt by TIE to use a complex instruction routine may result in a fatal memory access violation. This may happen because the complex instruction routines are in a separate image, TIE$EMULAT_TV.EXE, which is dynamically activated by LIB$FIND_IMAGE_SYMBOL on the first use of one of the routines. Depending on when this occurs and the address passed to the brk( ) call that releases memory, the memory into which TIE$EMULAT_TV.EXE is loaded may also be released.

  To avoid this problem, never use brk( ) to release memory, or be sure to execute a complex VAX instruction before getting the break address that is later used to release memory. Using brk( ) to allocate memory is fine.

- A translated VAX C program that uses vfork( ) and any executive function may hang at run time. If the child process of the VAX C program aborts erroneously, it may hang waiting for a mailbox I/O to be completed. One workaround is to prevent the child process from aborting.

4.29 Translated Image Support (Alpha Only)

V6.1

Alpha

At the beginning of the OpenVMS Alpha program, translation support was provided to remove impediments for users moving to Alpha due to:

- Lack of full language support initially
- Unavailability of source code for recompilation
- Difficulty of recompiling code that depended heavily on certain features of the VAX architecture

For languages whose VAX versions are undergoing active development, native Alpha versions are now available. The Translated Image Environment is being maintained to support those language features that were available as of the release of OpenVMS VAX Version 5.5-2.

Similarly, translation is supported for images whose use of system services and run-time library entry points is restricted to those that existed on OpenVMS VAX Version 5.5-2.

In cases where more recent VAX layered products have been installed, it may be necessary to take minor additional steps if application needs require rebuilding an image suitable for translation. For instance, with DECwindows Motif Version 1.2 or Version 1.2–3 for OpenVMS VAX, images must be built with the OSF Motif Version 1.1.3 library or the DECwindows XUI library rather than with the OSF Motif Version 1.2.2 or Version 1.2.3 library in order to be suitable for translation.

Similarly, for those using recent versions of DEC Fortran for VAX, an additional qualifier is required to compile Fortran programs that are suitable for translation.

For further information, see the release notes for particular VAX products.

As a safety measure for situations where future rebuilding and retranslation of OpenVMS VAX images is likely, it may be preferable to save copies of the relevant OpenVMS VAX Version 5.5-2 shareable images in a separate VAX directory and link new versions of VAX applications against those images. Using that technique the resulting images will be compatible with newer OpenVMS VAX shareable images (picking up any OpenVMS enhancements of existing features), and will also be properly built for translation to OpenVMS Alpha (by not requiring newer versions of shareable images).

The following sections list the translated images, image information files, and other related files that are provided with OpenVMS Alpha.

OpenVMS Alpha contains no translated message images. All message images have been made native.

Translated Images in SYS$LIBRARY:

BASRTL2_D53_TV.EXE
BASRTL2_D56_TV.EXE
BASRTL_D56_TV.EXE
BASRTL_TV_SUPPORT.EXE
BLAS1RTL_D53_TV.EXE
BLAS1RTL_D56_TV.EXE
COBRTL_D56_TV.EXE
DBLRTL_D56_TV.EXE
EDTSHR_TV.EXE
FORRTL2_TV.EXE
FORRTL_D56_TV.EXE
LIBRTL2_D56_TV.EXE
LIBRTL_D56_TV.EXE
MTHRTL_D53_TV.EXE
MTHRTL_D56_TV.EXE
PASRTL_D56_TV.EXE
PLIRTL_D56_TV.EXE
RPGRTL_TV.EXE
SCNRTL_TV.EXE
TECOSHR_TV.EXE
TIE$EMULAT_TV.EXE
UVMTHRTL_D53_TV.EXE
UVMTHRTL_D56_TV.EXE
VAXCRTLG_D56_TV.EXE
VAXCRTL_D56_TV.EXE
VMSRTL_TV.EXE

Translated Images in SYS$SYSTEM:

DBLMSGMGR_TV.EXE
EDF_TV.EXE
EDT_TV.EXE
MONITOR_TV.EXE
TECO32_TV.EXE

Translated RTL Images in IMAGELIB:

BASRTL2_D53_TV.EXE
BASRTL_D56_TV.EXE
BLAS1RTL_D53_TV.EXE
COBRTL_D56_TV.EXE
DBLRTL_D56_TV.EXE
FORRTL2_TV.EXE
FORRTL_D56_TV.EXE
LIBRTL_D56_TV.EXE
PLIRTL_D56_TV.EXE
RPGRTL_TV.EXE
SCNRTL_TV.EXE

Note that most of the translated RTLs are provided in D56 format rather than D53 format; some are provided in both formats. Where both formats are provided, the default format is D53. See Section 4.30 for more information about the translated run-time libraries.

Image Information Files in SYS$LIBRARY:

ACLEDTSHR.IIF
BASRTL2.IIF
BASRTL.IIF
BLAS1RTL.IIF
COBRTL.IIF
CONVSHR.IIF
CRFSHR.IIF
DBLRTL.IIF
DCXSHR.IIF
DISMNTSHR.IIF
DTKSHR.IIF
EDTSHR.IIF
ENCRYPSHR.IIF
EPC$SHR.IIF
FDLSHR.IIF
FORRTL.IIF
FORRTL2.IIF
INIT$SHR.IIF
LBRSHR.IIF
LIBRTL.IIF
LIBRTL2.IIF
MAILSHR.IIF
MOUNTSHR.IIF
MTHRTL.IIF
NCSSHR.IIF
P1_SPACE.IIF
PASRTL.IIF
PLIRTL.IIF
PPLRTL.IIF
PTD$SERVICES_SHR.IIF
RPGRTL.IIF
S0_SPACE.IIF
SCNRTL.IIF
SCRSHR.IIF
SECURESHR.IIF

SMBSRVSHR.IIF
SMGSHR.IIF
SORTSHR.IIF
SPISHR.IIF
TECOSHR.IIF
TPUSHR.IIF
UVMTHRTL.IIF
VAXCRTL.IIF
VAXCRTLG.IIF
VMSRTL.IIF

System Logical Names Definitions

The following system logical names are defined to facilitate the translated environment:

```
ACLEDTSHR_TV = ACLEDTSHR
CDDSHR_TV = CDDSHR
CONVSHR_TV = CONVSHR
CRFSHR_TV = CRFSHR
DCXSHR_TV = DCXSHR
DISMNTSHR_TV = DISMNTSHR
DTKSHR_TV = DTKSHR
ENCRYPSHR_TV = ENCRYPSHR
EPC$SHR_TV = EPC_SHR
FDLSHR_TV = FDLSHR
INIT$SHR_TV = INIT$SHR
LBRSHR_TV = LBRSHR
MAILSHR_TV = MAILSHR
MOUNTSHR_TV = MOUNTSHR
NCSSHR_TV = NCSSHR
PPLRTL_TV = PPLRTL
PTD$SERVICES_SHR_TV = PTD$SERVICES_SHR
SCRSHR_TV = SCRSHR
SECURESHR_TV = SECURESHR_JACKET
SMBSRVSHR_TV = SMBSRVSHR
SMGSHR_TV = SMGSHR
SORTSHR_TV = SORTSHR
SPISHR_TV = SPISHR
TPUSHR_TV = TPUSHR

BASRTL_TV = BASRTL_D56_TV
BASRTL2_TV = BASRTL2_D53_TV
BLAS1RTL_TV = BLAS1RTL_D53_TV
COBRTL_TV = COBRTL_D56_TV
DBLRTL_TV = DBLRTL_D56_TV
FORRTL_TV = FORRTL_D56_TV
LIBRTL_TV = LIBRTL_D56_TV
LIBRTL2_TV = LIBRTL2_D56_TV
MTHRTL_TV = MTHRTL_D53_TV
PASRTL_TV = PASRTL_D56_TV
PLIRTL_TV = PLIRTL_D56_TV
VAXCRTL_TV = VAXCRTL_D56_TV
VAXCRTLG_TV = VAXCRTLG_D56_TV
```

```
DBLMSGMGR = DBLMSGMGR_TV
EDTSHR_TV = EDTSHR
TECO32 = TECO32_TV
TECOSHR = TECOSHR_TV
VMSRTL = VMSRTL_TV

DBLRTLMSG = DBL$MSG
PASMSG = PAS$MSG
PLIMSG = PLI$MSG
RPGMSG = RPG$MSG
SCNMSG = SCN$MSG
VAXCMSG = DECC$MSG ♦
```

4.30 Translated Run-Time Libraries (Alpha Only)

V1.0

Alpha

As part of the OpenVMS Alpha kit, Digital provides a set of translated run-time libraries.

Some of the routines in the VAX run-time libraries use the VAX D_floating data type for double-precision arithmetic.

In the translated versions of these libraries, the Alpha D56 D_floating data type is used by default (where the VAX run-time library used D_floating). This provides the full precision of the 56-bit mantissa in VAX D_floating, yielding consistency of results at a cost in execution-time performance.

For a handful of performance-critical math-related libraries, Digital also supplies versions of the translated run-time libraries that use the Alpha D53 D_floating data type for double-precision operations. For these libraries, the D53 forms are the default. The D53 forms provide better performance by sacrificing the low-order three bits of precision in the mantissa.

The following translated libraries are provided in D56 form only:

- BASRTL
- COBRTL
- DBLRTL
- FORRTL
- LIBRTL
- LIBRTL2
- PASRTL
- PLIRTL
- VAXCRTL
- VAXCRTLG

The following translated libraries are provided in both D56 and D53 (the default) form:

- BASRTL2
- BLAS1RTL
- MTHRTL

- UVMTHRTL

Accessing the D56 Form of the Run-Time Libraries

V1.0

When you use the run-time libraries, the following happens by default:

- For BASRTL2, translated BASIC images that use MAT functions on double-precision data invoke BASIC run-time library routines that use the D53 data type.
- For BLAS1RTL, translated images that invoke BLAS$ functions with double-precision floating-point arguments get routines that use the D53 data type.
- For MTHRTL, translated images that invoke MTH$ double-precision floating-point functions get routines that use the D53 data type.
- For all others, the Alpha D56 floating-point data type is used by default.

Some users might need the full precision of D56 floating point. However, using the D56 routines imposes a very significant performance penalty. To access the D56 routines, redefine the run-time library's logical name to the D56 form, as shown in Table 4–11. The logical name can be defined on a per-process or systemwide basis, as appropriate for your site.

Table 4–11 Run-Time Library Logical Names

Library	Logical Name	D56 Name
BASRTL2	BASRTL2_TV	BASRTL2_D56_TV
BLAS1RTL	BLAS1RTL_TV	BLAS1RTL_D56_TV
MTHRTL	MTHRTL_TV	MTHRTL_D56_TV

4.30.1 Problems and Restrictions

The following note describes a problem with translated callers.

4.30.1.1 CRF$FREE_VM and CRF$GET_VM: Translated Callers Problem

V1.0

Translated callers to CRF$FREE_VM and CRF$GET_VM will not work properly. The translated callers are expecting VAX JSB semantics, but instead, Alpha JSB semantics are present in the native code (naturally).

To work around this problem, the translated callers need to use CALL instead of JSB. ♦

4.31 Translated VAX C Run-Time Library (Alpha Only)

Alpha

The following sections contain release notes pertaining to the translated VAX C run-time library.

4.31.1 Problems and Restrictions

This section describes known problems and restrictions with the OpenVMS Alpha translated VAX C Run-Time Library (VAX C RTL).

4.31.1.1 Functional Restrictions

V1.0 The translated VAX C RTL is a translated version of the OpenVMS VAX Version 5.4 VAX C RTL. All problems and restrictions present in that release of the VAX C RTL exist unchanged in the translated VAX C RTL. The following items are known restrictions in the functionality of the translated VAX C RTL:

- The fmod( ) function does not produce correct results for D_FLOAT.
- D_FLOAT programs that use the SIGFPE signal may not catch all floating-point exceptions. Translating the program using /FLOAT=D56_FLOAT fixes most SIGFPE problems.
- The sbrk( ) function returns an address that does not match the value returned from SYS$EXPREG.
- D_FLOAT programs that use the HUGE_VAL constant or call the math functions (which may return HUGE_VAL) may fail unless they are translated with /FLOAT=D56_FLOAT.
- Under certain circumstances, some math functions (either D_FLOAT or G_FLOAT) may generate a high-performance arithmetic trap exception instead of setting errno to ERANGE or EDOM.

4.31.1.2 Interoperability Restrictions

V1.0 The following restrictions apply when the translated VAX C RTL interoperates with the native DEC C RTL:

- The longjmp function cannot be used to transfer control from:
  - A native routine to a translated routine
  - A translated routine to a native routine
- Memory allocated by malloc, calloc, and so forth must be freed in the same context. That is, if a translated routine allocates memory, the free call must occur in a translated routine. Allocating memory in a translated routine and freeing it in a native routine results in corruption of the heap. Likewise, allocating memory in a native routine and freeing that memory in a translated routine also corrupts the heap.
- Signal handlers established by the signal (and related) functions in translated routines are not invoked when the signal is raised. Only native signal handlers can be used to catch UNIX style signals.
- The signals SIGEMT, SIGTRAP, SIGIOT and SIGFPE cannot be caught if those signals are raised by a translated image.
- The exec function can be used only to invoke similar images. That is, an exec function invoked in a native image cannot execute a translated image. Likewise, an exec function invoked in a translated image cannot execute a native image.
- An access violation occurs if vfork is executed in a native image to establish the context for a later system call and the system call is then invoked in a translated image.
- File pointers and file descriptors cannot be shared between native and translated images. An access violation or file corruption is likely to occur if a file is opened in a translated image and a native image attempts to read or write using that file pointer. The same results occur if a file is opened in a

native image and a translated image attempts to read or write using that file pointer.

Programs that perform any of these restricted actions may receive access violations or other exceptions. No testing is performed to detect and prevent restricted operations from being performed. ♦

4.32 Translated VAX COBOL Programs (Alpha Only)

V1.5

Alpha

The OpenVMS Alpha operating system supports the execution of translated VAX COBOL programs compiled with the VAX COBOL Version 5.0 compiler (or earlier compilers).

4.32.1 Problems and Restrictions

V1.5

Programs compiled with the VAX COBOL Version 5.1 compiler are not supported by the OpenVMS Alpha operating system.

4.33 VAX MACRO (VAX Only)

VAX

The following sections contain release notes pertaining to VAX MACRO.

4.33.1 Documentation Changes and Corrections

This section describes corrections to the VAX MACRO documentation.

4.33.1.1 *VAX MACRO and Instruction Set Reference Manual*

V6.1

The following sections describe corrections to the *VAX MACRO and Instruction Set Reference Manual*.

4.33.1.1.1 .ASCIZ Macro Directive

V6.1

Chapter 6 of the *VAX MACRO and Instruction Set Reference Manual*, Version 6.0 describes the .ASCIZ macro directive. In the example that illustrates the .ASCIZ directive, the following line of code is incorrect:

```
.ASCIZ /A/<KEY>(FF\TEXT)/B/       ; 3 characters in string,
                                  ;  4 bytes of data
```

Change this code to:

```
.ASCIZ /A/<FF>/B/         ; 3 characters in string
```

4.33.1.1.2 General Register Addressing

V6.1

Chapter 8 of the *VAX MACRO and Instruction Set Reference Manual*, Version 6.0 describes the general register addressing notations for the Macro assembler. In Table 8-5, the assembler notation for "Register deferred" (addressing mode 6) in Macro code is listed as:

Rn

Change to:

(Rn)

4.33.1.1.3 VAX Instruction CASE

V6.1 Chapter 9 of the *VAX MACRO and Instruction Set Reference Manual*, Version 6.0 describes the VAX instruction set. Where the description of the CASE instruction references symbol **disp1**[0] twice in the first note, the symbols erroneously use the digit 1 (one) instead of the letter "l".

Change both occurrences of **disp1**[0] to:

displ[0] ♦

4.34 X/Open Transport Interface (XTI)

The notes in this section describe the X/Open Transport Interface (XTI).

4.34.1 Changes and Enhancements

V6.2 OpenVMS Version 6.2 supports the X/Open Transport Interface (XTI) programming interface. The implementation conforms with the XPG4 X/Open CAE XO/CAE/91/600 (ISBN 1 872630 29 4) X/Open Transport Interface (XTI) specification.

Supported Transports

OpenVMS Version 6.2 supports the DECnet for OpenVMS (Phase IV) and TCP/IP transports. See Section 4.34.2 for support restrictions.

The transport names used in the t_open routine are 'dnet' for DECnet for OpenVMS, and either 'ip/udp' or 'ip/tcp' for TCP/IP.

Other transports are available with other networking layered products. Consult individual layered product documentation for information about OpenVMS XTI support.

Architecture

XTI is supported by front end and back end code. Front end code provides access to the standard interface routines. Back end code provides the interface from the front end code to the selected networking transport.

The supporting image files are as follows:

XTI front end code	SYS$SHARE:XTI$XTILIB.EXE
TCP/IP XTI back end code	SYS$SHARE:XTI$IPSHR.EXE
DECnet for OpenVMS XTI back end code	SYS$SHARE:XTI$DNETSHR.EXE
XTI C programming include file	SYS$LIBRARY:XTI.H

Linking Requirements

After compiling an XTI program, no additional qualifiers are required for linking with XTI.

Documentation

Documentation about XTI is not included in the OpenVMS Version 6.2 release. You can order documentation directly from X/Open Company Limited. If you have access to the Internet, you can get more information about X/Open Company Limited (including their publications) by browsing the following URL:

http://www.xopen.co.uk/

You can also contact X/Open Company Limited at the following locations:

- USA: East Coast

  X/Open Company Limited
  3141 Fairview Park Drive
  Falls Church
  VA 22042-4501
  U.S.A.
  Tel: +1 (703) 876 0044
  Fax: +1 (703) 876 0050

- USA: West Coast

  X/Open Company Limited
  1010 El Camino Real
  Suite 380
  Menlo Park, CA 94025
  U.S.A.
  Tel: +1 (415) 323 7992
  Fax: +1 (415) 323 8204

- United Kingdom:

  X/Open Company Limited
  Apex Plaza
  Forbury Road
  Reading
  Berks RG1 1AX
  U.K.
  Tel: +44 1734 508311
  Fax: +44 1734 500110

- Japan:

  X/Open Company Limited
  Karufuru-Kanda Bldg, 9F
  1-2-1 Kanda Suda-cho
  Chiyoda-Ku
  Tokyo 101
  Japan
  Tel: +81 3 3251 8321
  Fax: +81 3 3251 8376

4.34.2 Problems and Restrictions

V6.2

The following restrictions apply to the OpenVMS Version 6.2 XTI:

- Nonblocking I/O is unsupported for DECnet for OpenVMS.

  DECnet for OpenVMS (Phase IV) does not fit the model for nonblocking I/O. Attempts to open or switch an XTI file descriptor to nonblocking I/O (O_NONBLOCK) will fail.

- Connectionless I/O is unsupported for DECnet for OpenVMS.

  DECnet for OpenVMS (Phase IV) does not fit the model of connectionless I/O. Therefore, only connection-oriented connections are supported.

- Disabled ASTs cause problems.

  The XTI back end code uses ASTs for asynchronous delivery of events from the transports. If ASTs are disabled (sys$setast(0)), the XTI back end code will not operate correctly until the ASTs are enabled again.

- XTI file descriptors are not compatible with C Run-Time Library file descriptors.

In addition, the 't_info' structure returned from the t_open function reports any additional implementation-specific restrictions for the given transport. (See the XTI documentation for information about the t_open command.)

A

ISA Bus Support (Alpha Only)

Alpha

Alpha platforms with Peripheral Component Interconnect (PCI) and Industry Standard Architecture (ISA) I/O bus support use industry standard bus components to implement the ISA bus interface. For this reason, the Alpha platforms provide the same limited set of bus resources for ISA options that are available on a PC. These bus resources are IRQ lines, DMA channels, I/O ports, and ISA memory buffers.

To allocate the bus resources to the various ISA devices that require them, some kind of resource management is required. On ISA machines, the console provides an ISACFG command that allows a user to enter ISA configuration data into the system NVRAM. ISA bus support and bus configuration routines then extract the ISA configuration data from the console NVRAM. The console NVRAM ISA configuration data is referred to as the console **ISA configuration table**.

The console ISA configuration table contains resource usage information for each of the ISA devices in the system. An additional mechanism interprets the table entries and loads the correct drivers for the devices that are present. The OpenVMS **SYS$MANAGER:ISA_CONFIG.DAT** file stores driver and device name information. A user can edit this file to enter ISA bus configuration information as well as driver and device names. OpenVMS ISA bus configuration routines use the console ISA configuration table as a source of ISA bus resource usage and use the ISA_CONFIG.DAT file as a source of driver loading information, which includes driver and device names.

This chapter describes ISA device configuration on machines that support ISA as an I/O bus, and it contains procedures for configuring an ISA device. It also provides an example SYS$MANAGER:ISA_CONFIG.DAT file. For information about configuring ISA cards on an EISA bus, see *OpenVMS AXP Device Support: Developer's Guide*.

A.1 OpenVMS ISA Bus Configuration

The basic strategy for ISA bus configuration in OpenVMS is to configure the devices found in the console ISA configuration table, and then configure the devices found in the ISA_CONFIG.DAT file.

During booting, the I/O bus mapping routine INI$IOMAP probes the console ISA configuration table and copies bus configuration information from the console ISA configuration table to the ISA ADP/Busarray. During autoconfiguration, IOGEN$ISA_CONFIG walks the ADP/Busarray and configures the devices found in the Busarray. IOGEN$ISA_CONFIG then opens file ISA_CONFIG.DAT and configures the devices found in the file.

A.2 Adding a Device

A machine fresh from the factory contains a console ISA configuration table with entries for all of the built-in devices (keyboard, mouse, com1, com2, lpt1, and floppy) and entries for any factory installed ISA options. Each console ISA configuration table entry lists the bus resources consumed by that device, as well as a 16 byte HANDLE (an ASCII character string), that is used to identify the device. OpenVMS will configure all factory installed and built-in devices using data from the console ISA configuration table. OpenVMS uses the HANDLE field of each console ISA configuration table entry to associate a device and a driver.

To configure an add-in ISA option device, you must first enter the IRQ used by the device in the console ISA configuration table as described in Section A.2.1.

Note

This step is done prior to booting the operating system, so that the console knows that a particular IRQ is in use by an ISA device. This will prevent the console from assigning this IRQ to a PCI device. If you do not perform this step, the system might become unresponsive ("hang") because two devices are trying to use the same IRQ.

After you boot the operating system, you can load a driver for an ISA device in one of two ways:

- Using ISA_CONFIG.DAT file
- Using SYSMAN

See Section A.2.2 and Section A.2.3 for more information about both of these methods.

A.2.1 Entering Interrupt Request Line (IRQ) Assignments

As mentioned in Section A.2, before you add optional devices to an ISA bus, you must enter the IRQ used by the device in the console ISA configuration table using the ISACFG command as described in this section. For additional information about the ISACFG command, see the *AlphaStation 400 Series User Information* manual. The SRM console on your Alpha system assigns interrupt vectors to the PCI devices plugged into the PCI option slots. The console automatically assigns an available interrupt request line (IRQ) to each PCI device based on the configuration information stored by the console configuration utility with the system.

To avoid a conflict where an IRQ that you want to use for an ISA device is already reserved for a PCI device, use the console configuration utility to reserve the IRQ for the ISA device. As noted in Section A.2, these conflicts might cause the system to become unresponsive because two devices are trying to use the same IRQ.

The commands you enter at the console prompt (>>>) are similar to the following:

```
>>> isacfg -mk -etyp 1 -enadev 1 -dev 0 -slot x -irq0 y
>>> init
```

Note the following conventions:

- *x* is the ISA option slot into which you insert the card.

For more information about ISA option slots, see the NODE=x description in Section A.2.2.

- *y* is the IRQ that you are reserving for your device.

Note the following as well:

- To verify that your command correctly reserved the IRQ, enter the following command:

  ```
  >>> isacfg -all
  ```

  The system will display the entire data structure of ISA configuration information stored in the console, including the IRQ you just reserved.

- If you made an error, you can enter the following command to cancel the assignment:

  ```
  >>> isacfg -rm -dev 0 -slot x
  ```

- If you are using your system as a Universal Platform booting the OSF/1 operating system, you may have already made an entry using the console configuration utility. You do not need to enter the information again. (If you do, the system will display a message informing you that the slot is already reserved for an existing ISA device.)

  If you want to override the existing information enter the following command:

  ```
  >>> isacfg -mod -slot x -dev 0 -etyp 1 -enadev 1 -irq0 y
  ```

  Note the following conventions:

  - *x* is the ISA option slot into which you insert the card.
  - *y* is the IRQ assigned to the device.

  Note that this specific command does not affect other configuration information values stored for the device; the command modifies only the IRQ information.

A.2.2 Configuring a Device with an ISA_CONFIG.DAT File

If you are using the ISA_CONFIG.DAT file to configure your device, you must first copy SYS$MANAGER:ISA_CONFIG.TEMPLATE to SYS$SPECIFIC:[SYSMGR]ISA_CONFIG.DAT.

Then edit the ISA_CONFIG.DAT file to specify the device name, driver name, and other ISA bus resource usage (such as DMA channels, I/O ports, and memory buffers) for your device. (This is analogous to editing config.sys on a PC.) You must enter the IRQ for your device in ISA_CONFIG.DAT, even though you already entered the IRQ in the console ISA configuration table (as described in Section A.2.1).

The main advantage of using ISA_CONFIG.DAT for add-in ISA option configurations is that the device is automatically configured. There is no need to write an ICBM or to use the SYSMAN IO CONNECT command, even for devices that OpenVMS does not support.

The ISA_CONFIG.TEMPLATE file is shipped in the system manager's directory with examples of configuration entries for many ISA cards. The file is shipped with all entries commented out. An example entry from the SYS$MANAGER:ISA_CONFIG.TEMPLATE file is as follows:

```
;       Example 1: To indicate that the DE202 Card is plugged into slot 1,
;       using IRQ 5, I/O ports 300-30F, and memory buffer D0000-DFFFF, enter
;       the following into this SYS$SPECIFIC:[SYSMGR]ISA_CONFIG.DAT.
;
[ERA0]1
NAME=ER2                  ; device name
DRIVER=SYS$ERDRIVER3      ; driver name
NODE=14                   ; plugged into ISA Option slot 1
IRQ=55                    ; device is using irq 5
PORT=(300:f)6             ; 16 bytes starting at 300
MEM=(D0000:10000)7        ; 64 Kbytes starting at D0000
USER_PARAM = thinwire8    ; optional user parameter.  Available to
                                ; driver via IOC$NODE_DATA call using function
                                  code of IOC$K_ISA_USER_PARAM=18.
```

1 The opening bracket indicates that the following lines contain ISA configuration information. This is basically an entry delimiter. The text between the brackets is not used for anything.

2 The NAME=xx field allows specification of a standard two letter VMS device name. This two-letter string, along with a controller letter and a unit number, is passed to the driver loading service as a standard OpenVMS device name. The two-letter device string is also copied directly to the busarray$q_hw_id field in the ISA bus array entry specified by the NODE field. This will cause the SYSMAN IO SHOW BUS command to display this device as "Unknown ISA," instead of not displaying it.

3 The DRIVER=drivername specifies the driver that should be loaded for this device. This name is limited to 16 ASCII characters, not including the implicit .EXE.

4 The NODE=x field is used by OpenVMS to keep track of where ISA configuration information is being stored in the operating system data structures. Note that any correspondence with the actual location of the device in a physical bus slot is not needed. However, if you are configuring more than one ISA device in the ISA_CONFIG.DAT file, you must use unique node numbers to avoid confusing the OpenVMS ISA bus configuration code. It might be helpful to label the ISA option slots 1, 2, 3, 4 with a permanent felt-tip marker starting with the slot on the bottom (assuming an upright AlphaStation 400 Series system). Then use these slot numbers for the NODE=x portion of the ISA_CONFIG.DAT entries. When you enter the IRQ for your device at the console with the ISACFG command, the slot number that you specify must match the node number specified in the entry for the device in the ISA_CONFIG.DAT file.

5 The IRQ=x field tells OpenVMS ISA configuration code which IRQ is being used by the device. OpenVMS ISA configuration code derives the interrupt vector that will be used by the driver interrupt service routine from the ISA IRQ.

Even though you specified the IRQ at the console with the ISACFG command, you also must specify it here.

On AlphaStation 200 and 400 Series systems, the built-in devices use many of the available ISA IRQ lines as shown in Table A–1.

Table A–1 Available ISA IRQ Lines

IRQ	Device
0	TIMER (built-in)
1	KBD (built-in)
2	Dual 8259 cascade
3	COM2 (built-in)
4	COM1 (built-in)
5	Available for PCI or ISA option slot
6	FLOPPY (built-in)
7	LPT1 (built-in)
8	Interval timer (not used on these systems)
9	SOUND (built in device on AlphaStation 200 only), available for PCI or ISA option slot on AlphaStation 400
10	Available for PCI or ISA option slot
11	PCI NCR810 SCSI (built-in)
12	MOUSE (built-in)
13	DMA buffer chaining (not available outside PCI/ISA bridge)
14	Available for PCI or ISA option slot
15	Available for PCI or ISA option slot

Note that only IRQs 5, 9, 10, 14, and 15 are available for add-in devices on AlphaStation 400 Series systems. Only IRQs 5, 10, 14, and 15 are available on AlphaStation 200 Series systems because the built-in sound card uses IRQ 9. These are the only IRQ lines available for all of the PCI and ISA option slots.

You must be careful not to specify an IRQ that is in use by another device because there is currently no IRQ sharing in OpenVMS. Only one driver interrupt service routine can be connected to an IRQ. If you mistakenly overload an IRQ, only one driver (the one that is loaded last) will ever get interrupts from that IRQ.

6 The PORT=(x:y) field allows specification of the I/O port usage of the option. Up to 9 I/O port ranges can be listed:

```
;   PORT=(aa:b,cc:d,...)     ; where aa and cc are the ISA I/O port base
;                            ; addresses for the device, and b,d indicate
;                            ; the number of bytes to be reserved.
;                            ; Up to 9 separate I/O port ranges can be
;                            ; specified.
```

If a driver needs to know which I/O ports its device is using, the driver can call the IOC$NODE_DATA routine to get the I/O port information. See Section A.3 for more information about the IOC$NODE_DATA routine.

7 The MEM=(x:y) field allows specification of the ISA memory buffers used by the ISA device. Up to 4 memory buffers can be specified:

```
;   MEM=(ee:f,gg:h,...)      ; where ee and gg indicate the base ISA memory
;                            ; address for the device's on-card buffer, and
;                            ; f,h indicate the size of the buffer in bytes.
;                            ; Up to 4 different memory buffer address
;                            ; ranges can be specified.
```

8 The USER_PARAM= field allows the user to create a string (72 bytes maximum) that is then available to the driver by way of the IOC$NODE_ DATA routine. This is intended to be a general mechanism for passing user defined information to a driver.

This particular example did not specifiy a DMA channel, because this ISA card does not use DMA channels. To specify a DMA channel, use the following:

```
DMA=(j,k,...)                   ; where j and k are values (0-7) that
                                ; specify which channel(s) of the DMA
                                ; controller this device is using to
                                ; relay information.
```

You may specify up to 4 DMA channels for your device.

After you have edited the ISA_CONFIG.DAT file, reboot the machine. Your device should be automatically configured, and the driver should be loaded.

A.2.3 Configuring an ISA Device with SYSMAN

To manually configure your ISA device, you can also use the System Management (SYSMAN) utility instead of using ISA_CONFIG.DAT file. The following example shows how to to configure your device using the SYSMAN IO CONNECT command:

To use the SYSMAN utility, enter the following command:

```
$ MCR SYSMAN
```

At the SYSMAN> prompt, enter the following IO CONNECT command:

```
SYSMAN> IO CONNECT devname /ADAPTER=x /DRIVER=drivername -
/NODE=slot /CSR=baseaddress /VECTOR=irq*4
```

The arguments and qualifiers to the IO CONNECT command are as follows:

devname
Specifies the OpenVMS device name of your device. This should be specified as a standard OpenVMS device name–a 2 letter device code, a controller letter, and a unit number.

/ADAPTER=x
Specifies the OpenVMS software adapter number of the ISA bus. Before you enter the IO CONNECT command, you should display the I/O adapters that are known to OpenVMS by entering the following command:

```
SYSMAN> IO SHOW BUS
```

The I/O adapter numbers are the "TR#" from the IO SHOW BUS DISPLAY command. The following example shows sample output from the SYSMAN IO SHOW BUS command on an AlphaStation 400 Series system:

```
SYSMAN> io show bus
```

%SYSMAN-I-OUTPUT, command execution on node JIMMYJ

```
_Bus_________Node_TR#__Name___________Base CSR_________
  PCI          48   2    PCI/SCSI       FFFFFFFF822DC000
      XBUS          0    4    Mouse          FFFFFFFF820DC000
      XBUS          1    4    Keyboard       FFFFFFFF820DC000
      XBUS          2    4    Serial Port    FFFFFFFF820DC000
      XBUS          3    4    Serial Port    FFFFFFFF820DC000
      XBUS          4    4    Parallel Port  FFFFFFFF820DC000
      XBUS          5    4    Floppy         FFFFFFFF820DC000
    ISA           1    3    Lance NI Adapt FFFFFFFF820DC000
    ISA           2    3    Sound Adapter  FFFFFFFF820DC000
SYSMAN>
```

In this display, you can see that the ISA bus adapter has TR# 3. Because you are plugging a device in to the ISA bus, you should specify /ADAPTER=3 in the SYSMAN IO CONNECT command. This will cause your driver data structures to be associated with the ISA bus data structures.

Note that if your system has no devices plugged in to the ISA bus, then there will be nothing displayed for the ISA bus when you issue the SYSMAN IO SHOW BUS command. You will not see any entries for ISA bus devices. This is a known deficiency in the SYSMAN IO SHOW BUS command. However, you will always see the XBUS devices in the SYSMAN IO SHOW BUS display. The TR# for the ISA bus will always be one less than the TR# for the XBUS. For the AlphaStation 200 and 400 Series systems, the ISA adapter number will be 3, as shown in the previous example.

/DRIVER=drivername

The DRIVER=drivername qualifier specifies the file name of the driver that is to be loaded.

/NODE=slot

The /NODE=slot qualifier identifies the OpenVMS ISA bus array entry that will be used to store configuration information about your ISA device. Note that this need not have any correspondence with the physical ISA slot cutouts on the back of the machine. It is necessary, however, that you do not configure more than one device into the same slot. In the previous example SYSMAN IO SHOW BUS display, slots 1 and 2 are used by the Lance NI Adapt and the Sound Adapter. You must not use either of these slot numbers. It is also necessary that the slot argument match the slot number that you specified when you entered the console ISACFG command to reserve the IRQ. As mentioned earlier, it is suggested that you label the ISA bus slots and use those slot numbers in the console ISACFG command, in the ISA_CONFIG.DAT file (if you decide to configure your device using this file), and in the /NODE=slot qualifier in the SYSMAN IO CONNECT command (if you decide to configure your device this way).

/CSR=baseaddress

The driver loading service that is called by the SYSMAN IO CONNECT command requires the /CSR=baseaddress input qualifier. The driver loading service copies the baseaddress from the /CSR=baseaddress qualifier to the IDB$Q_CSR field in the IDB. The Base CSR in the SYSMAN IO SHOW BUS display for the XBUS devices is the base virtual address of ISA I/O space. The base address of ISA I/O space is generally a useful value for ISA device drivers and should be the value specified in the /CSR=baseaddress argument.

Note that if you code your driver using the IOC$MAP_IO routine to map your device register space, you may not need the baseaddress from the /CSR=baseaddress qualifier. However, you still must supply something for the /CSR=baseaddress qualifier, because it is required by the driver loading

service. If you are using IOC$MAP_IO and you are not depending on any base address value being passed to you by the driver loading service, you should say /CSR=0 for this argument.

/VECTOR=irq*4
The /VECTOR=irq*4 qualifier specifies to the driver loading service the software interrupt vector in use by your device. Your driver interrupt service routine is connected to the software interrupt vector and will be invoked when your device generates a hardware interrupt. The irq value in this qualifier must be the same as the IRQ value that you specified when you entered the console ISACFG command to reserve the ISA irq.

A.3 ISA Bus Device Driver Routines

This section describes how to use the IOC$NODE_DATA and IOC$NODE_ FUNCTION routines for a device driver using an ISA bus.

The bus support routine IOC$NODE_DATA is called by a driver to get bus slot/platform-specific information. The uses and capabilities of this routine vary from bus to bus. The bus support routine IOC$NODE_FUNCTION is called to perform bus slot-specific actions on behalf of a driver, such as enabling interrupts for a bus slot.

This section contains the prototypes for the IOC$NODE_DATA and IOC$NODE_ FUNCTION routines. For more information about using these routines, see the *OpenVMS AXP Device Support: Reference* manual.

The formats of the IOC$NODE_DATA and IOC$NODE_FUNCTION routines are as follows:

```
int ioc$node_data (CRB  *crb, int  function_code, void *user_buffer)

int ioc$node_function (CRB  *crb, int  function_code)
```

Both of these routines use the CRB$L_NODE field from the CRB to find the ADP and ISA bus array entry for the device. The supported function codes for IOC$NODE_DATA on the ISA bus are as follows:

IOC$K_EISA_IRQ	Return IRQ used by device
IOC$K_EISA_DMA_CHAN	Return DMA channel used by device
IOC$K_EISA_MEM_CONFIG	Return memory buffer(s) used by device
IOC$K_EISA_IO_PORT	Return I/O port(s) used by device

On the ISA bus, IOC$NODE_DATA uses the CRB$L_NODE field from the CRB to find the ADP and the ISA bus array entry for the device. The ISA bus array entry contains a pointer to a data structure called an ISA_CFG_DATA (defined by isacfgdef.h from sys$lib_c.tlb) to find the requested information. The contents of the ISA_CFG_DATA are derived from two places. If you are configuring your device using the ISA_CONFIG.DAT file, the information returned by IOC$NODE_ DATA is derived from the entry that you made in ISA_CONFIG.DAT for your device. If you are configuring your device using the SYSMAN IO CONNECT command, there should not be an entry in the ISA_CONFIG.DAT file. In this case the information returned by IOC$NODE_DATA is derived from the entry for your device in the console ISA configuration table, which you created when you used the console ISACFG command to reserve the IRQ for your device.

When calling IOC$NODE_DATA, you must supply a user buffer large enough to contain the information. The data size returned for each of the previously mentioned function codes is as follows:

IOC$K_EISA_IRQ	The IRQ used by the device is returned as a 32 bit int.
IOC$K_EISA_DMA_CHAN	The DMA channel number assigned to the device is returned as a 32 bit int. If more than one DMA channel is specified in ISA_CONFIG.DAT, then each one is returned as an int (an array of ints should be declared to receive the information).
IOC$K_EISA_MEM_CONFIG	The memory buffer used by the device is returned in an array of two consecutive 32 bit ints. The first array element contains the ISA bus base memory buffer address, and the second array element contains the memory buffer size (in bytes). If more than one memory buffer is specified in ISA_CONFIG.DAT, each one is returned in an array of ints. You should declare an array of ints that is big enough to contain all of the memory buffer information used by your device.
IOC$K_EISA_IO_PORT	The I/O port information used by the device is returned as 32 bit integers. The low 16 bits of the 32 bit integer contain the base I/O port, and the upper 16 bits contain the number of consecutive I/O ports from the base I/O port. If more than one I/O port is specified in ISA_CONFIG.DAT, each one is returned as a 32 bit int. You should declare an array of ints large enough to contain all of the I/O port information used by your device.

The supported function codes for IOC$NODE_FUNCTION on the ISA bus as follows:

IOC$K_ENABLE_INTR	Enables hardware device interrupt in the ISA interrupt control logic
IOC$K_DISABLE_INTR	Disables hardware device interrupt in the ISA interrupt control logic

IOC$NODE_FUNCTION also uses the CRB$L_NODE field from the CRB to find the ADP and ISA bus array entry for the device. IOC$NODE_FUNCTION then finds the IRQ used by the device and either enables (if called with the IOC$K_ENABLE_INTR function code) or disables (if called with the IOC$K_DISABLE_INTR function code) the IRQ line used by the device.

A.4 Determining an Available ISA IRQ

To determine which IRQs are available for use by an ISA device, you must check a few things. The console utility will have IRQs reserved for system board devices, for ISA Option slot devices that have been entered by the user, and for any PCI Options that have been installed.

The file SYS$MANAGER:ISA_CONFIG.DAT may contain resource information for ISA option devices that the console does not know about.

In an AlphaStation 200 Series, the only available IRQs for ISA and PCI Option devices are 5, 10, 14, 15. In the AlphaStation 400 Series, the Audio option is not present, so the available IRQ list is 5, 9, 10, 14, 15. Note that thare are more option slots available than there are IRQs.

To determine if any of these IRQs are already consumed by devices previously installed in the system (either at the factory or by the user), you can follow this procedure:

1. Issue the console command ISACFG -ALL. This will output a list of all the ISA devices (and their resources) that the console knows about, including system board devices and any option slot devices that the user has entered. Search the output from this command to see if any of the IRQs from the available list are consumed by previously installed devices.

2. PCI Option slot devices report their interrupts through ISA IRQs also. The console is responsible for assigning these IRQs. It does so in a linear fashion, starting with the lowest numbered IRQ that is not specified as reserved in the console ISA configuration table. Each consecutive PCI slot is assigned the lowest available IRQ until all PCI options are assigned or we run out of IRQs. If there are no ISA Option slot device entries in the console ISA configuration table, the PCI Options would be assigned IRQs in order from the available list. If there is an entry for an ISA option slot device in the table, specifying IRQ 9, then any PCI Option slot devices would be assigned IRQs from the list 5,10,14,15. (Audio is a good example of this.)

   Use the console SHOW CONFIG command to list all the PCI devices. Note that the Intel SIO Bridge does not require an interrupt, and that the NCR 53C810 SCSI chip is hardwired to IRQ 11. Any other PCI devices that show up will consume IRQs from the available list in the manner previously described.

3. Boot the system and look through the file SYS$MANAGER:ISA_CONFIG.DAT for valid entries (not commented out). They will be specifying an IRQ. (Note that there should already be an entry reserving this IRQ in the console ISA configuration table if proper procedure is followed). If you find an IRQ listed that is not already specified in the console data, make an entry using the console ISACFG command as described earlier in this chapter.

   Any remaining IRQs are available for use by a new ISA Option card. Please note that you must reserve the IRQ using the console to ensure that an IRQ is reserved for your device.

A.5 Troubleshooting

This section contains information that might help you to solve some common problems that you might encounter while configuring your ISA device. If you have trouble adding a new device to the system, you can check a few things. More than likely, the device you are trying to add is conflicting with an existing device's ISA Resource assignments. For possible resource conflicts, check the output from the console's ISACFG -ALL command and compare with the contents of your SYS$MANAGER:ISA_CONFIG.DAT file.

Possible types of conflicts and suggested solutions include the following:

ISA IRQ conflicts

As mentioned in Section A.4, on AlphaStation 200 and 400 Series systems there are more option slots than there are IRQs.

ISA I/O port, DMA channel, and memory buffer conflicts

Choose a new resource for your board, or adjust the existing boards resource so that the needed one is available.

Slot numbering conflicts

ISA does not have slot numbers. OpenVMS requires some way to keep track of devices in the ISA bus and uses slot number as a convenient method. If there are slot number conflicts in the ISA_CONFIG.DAT file, the OpenVMS code will overwrite the first driver's resource information with the last driver's information and fail to load the first driver.

You must also ensure that there are no slot number conflicts between the console data and the ISA_CONFIG.DAT data.

Valid ISA slot numbers for the AlphaStation Series 200/400 are 1, 2, 3, and 4.

If you made changes to your SYS$MANAGER:ISA_CONFIG.DAT file and can no longer boot, you should boot conversationally using the following command. (Remember to re-enable autoconfiguration after the problem has been resolved.)

```
>>> boot -flags 0,1 <boot_device>

SYSBOOT> SET NOAUTOCONFIGURATION 1
SYSBOOT> CONTINUE
```

This will boot OpenVMS without configuring any of your I/O devices. You can then edit the SYS$MANAGER:ISA_CONFIG.DAT file to correct any problems.

If you made changes to the console data and are prevented from booting, use the console ISACFG command to remove the changes that you made.

A.6 System Board Resources for AlphaStation 200 and 400 Series Systems

System board resources for AlphaStation 200 and 400 Series systems are as follows:

COM1:	IRQ=4, I/O port=3F8:8
COM2:	IRQ=3, I/O port=2F8:8
LPT1:	IRQ=7, I/O Port=378:8
FDC:	IRQ=6, I/O port=3F0:8, DMA chan=2
Mouse:	IRQ=C, I/O port=60,64
Keyboard:	IRQ=1, I/O port=60,64
TOY Clock:	I/O port=70,71
PCI NCR810 SCSI:	IRQ=B, I/O port=26,27
Sound:	(This is a system board device on AlphaStation 200 Series only) IRQ=9, I/O Ports=388:4,530:8, DMA chan=(0,1)
Timer /Counter:	IRQ=0

INTEL SIO PCI /ISA Bridge interrupt logic:	IRQ=2 (Cascade IRQ for the dual 8259 on the bridge chip) IRQ=D (used for DMA Buffer Chaining, unconnnected on these systems) IRQ=8 (used for interval timer, unconnected on these systems) DMA=4 (used by the DMA controller as the cascade line)

A.7 Sample ISA_CONFIG.DAT File

This section contains a sample ISA_CONFIG.DAT file that is similar to the file you can edit.

```
;
;       I S A _ C O N F I G . D A T
;
;       This file informs the OpenVMS Alpha operating system which
;       devices are connected to the ISA bus.
;
;       Note: Do NOT make changes in this file.
;       To add option devices to this file you must first copy
;       this template file (SYS$MANAGER:ISA_CONFIG.TEMPLATE)
;       to SYS$SYSROOT:[SYSMGR]ISA_CONFIG.DAT.  Then edit the file
;       SYS$SYSROOT:[SYSMGR]ISA_CONFIG.DAT to add devices.
;
;   Contents of this file:
;   ----------------------
;       o Description of configuration command sets
;
;       o Example configuration command sets for supported ISA option cards
;
;       o System board resources for AlphaStation 200 and 400 series systems
;
;
;
;   Conventions used in this file:
;   ------------------------------
;       o Characters following a semi-colon (;) are comments.
;
;       o All numbers must be specified in hexadecimal.
;
;       o You must separate records in the file by using square brackets
;         around each ISA device that is specified (for example, [xyzn]).
;
;       o Spaces are ignored.
;
;       o Note the following about keywords:
;
;         - Keywords can be in any order, *except* in the following
;           instance: the NAME keyword must precede the NODE keyword or
;           the SYSMAN IO SHOW BUS command will not include the ISA
;           devices in the display.
;
;         - Each keyword must be on its own line.
;
;         - All keywords required for a device must be included before
;           the next record [xyzn] begins.
;
;   Description of Configuration Command Sets:
;   -------------------------------------------
;
;       Specify the following information for each ISA device. Note
;       that the NAME, DRIVER, NODE, and IRQ fields are required.
;
```

```
;       Note that fields that are not used must be omitted. For example, if
;       the device does not use a DMA channel, you must omit the DMA parameter.
;
;       NAME=xx                 ; where xx is the 2-letter device code,
;                               ; (for example: ER for the DE202)
;
;       DRIVER=driver_name      ; The name of the driver stored in
;                               ; SYS$LOADABLE_IMAGES.
;                               ; For example: SYS$ERDRIVER for the DE202, and
;                               ; SYS$IRDRIVER for the Proteon Token Ring card.
;
;       IRQ=i                   ; where i is a value (0-F) that specifies which
;                               ; ISA IRQ the device uses to report interrupts.
;
;       NODE=n                  ; where n is the ISA option slot number
;                               ; the device is plugged into.
;
;       DMA=(j,k,...)           ; where j and k are values (0-7) that
;                               ; specify which channel(s) of the DMA
;                               ; controller this device is using to
;                               ; relay information.
;                               ;
;                               ; Note the comma between indicators and the
;                               ; parentheses that indicates a list
;                               ; of parameters.
;                               ;
;                               ; Up to 4 DMA channels can be specified.
;
;       PORT=(aa:b,cc:d,...)    ; where aa and cc are the ISA I/O port base
;                               ; addresses for the device, and b,d indicate
;                               ; the number of bytes to be reserved.
;                               ; Up to 9 separate I/O port ranges can be
;                               ; specified.
;
;       MEM=(ee:f,gg:h,...)     ; where ee and gg indicate the base ISA memory
;                               ; address for the device's on-card buffer, and
;                               ; f,h indicate the size of the buffer in bytes.
;                               ; Up to 4 different memory buffer address
;                               ; ranges can be specified.
;
;       FLAGS = n               ; Currently, bit 1 is the only flag in use.
;                               ; It indicates that the device being configured
;                               ; is a SCSI adapter. Therefore, if the adapter
;                               ; is a SCSI  device, set this bit to 1. This
;                               ; will allow IOGEN$SCSI_CONFIG (which probes for
;                               ; devices on the SCSI bus) to locate the devices.
;
;       USER_PARAM = text       ; This parameter is dedicated solely for
;                               ; driver writer use.  A pointer to a copy of
;                               ; text is passed back in the user_buffer
;                               ; parameter from a call to IOC$NODE_DATA
;                               ; using a function code of IOC$K_ISA_USER_PARAM
;                               ; (note that the text is limited to 72 chars)
;
;
;
;       ************************* IMPORTANT ******************************
;
;       o For all of the following examples, be sure to verify that the
;         switches and jumpers on the card being inserted are set to
;         match the specified parameters BEFORE you remove the comment
;         characters (;) from this file.
;
;       o Be sure that there are no conflicting resource assignments between
;         devices (including the built-in devices on the system board and the
```

```
;         PCI option slot device IRQ assignments.)
;
;       o Be sure to copy SYS$MANAGER:ISA_CONFIG.TEMPLATE to
;         SYS$SYSROOT:[SYSMGR]ISA_CONFIG.DAT before making changes.
;
;         A list of resources assigned to system boards follows the examples.
;
;       ************************************************************************
;
;
;   Example Configuration Command Sets for Supported ISA Option Cards
;   ------------------------------------------------------------------
;
;       Following are examples of configuration command sets necessary to
;       automatically configure option cards that are not shipped with
;       the system.
;
;       Example 1: To indicate that the DE202 Card is plugged into slot 1,
;       using IRQ 5, I/O ports 300-30F, and memory buffer D0000-DFFFF, enter
;       the following into this file:
;
;       [ERA0]
;       NAME=ER
;       DRIVER=SYS$ERDRIVER
;       NODE=1                  ; plugged into ISA Option slot 1
;       IRQ=5
;       PORT=(300:f)            ; 16 bytes starting at 300
;       MEM=(D0000:10000)       ; 64Kbytes starting at D0000
;       USER_PARAM = thinwire   ; optional user parameter is passed directly
;                               ; back to driver
;
;       Example 2: To configure the DigiBoard PC/8 card, remove the comment
;       characters (;)  from the following. (Be sure to verify that the
;       switches are set to match the indicated resources, or change the
;       resource lists to match the switches.)  Note that SYS$YSDRIVER
;       expects the Master Status register address to be listed first.
;
;       [TTA0]
;       NAME=TT
;       DRIVER=SYS$YSDRIVER
;       IRQ=5
;       NODE=2                  ; plugged into ISA Option slot 2
;       PORT=(390:8,398:8,3a0:8,3a8:8,3b0:8,3b8:8,3c0:8,3c8:8,3d0:8)
;
;       Note that this configures only a single port on the card.  The
;       remaining 7 ports each need to be connected manually using the
;       following command:
;
;       $ MCR SYSMAN IO CONNECT/NOADAP/DRIVER=SYS$YSDRIVER TTAx:
;       ; where x is the unit number
;
;
;       Example 3: To configure a Microsoft Sound card, remove the
;       comment characters (;) from the following.  These commands
;       specify the card in ISA Bus Option slot 3, running with IRQ 9,
;       using DMA channels 0 and 1, and I/O Port addresses 388-38B,530-537.
;
;       [AUA0]
;        NAME=AU
;        NODE=3
;        DRIVER=SYS$MSBDRIVER
;        IRQ=9
;        DMA=(0,1)
;        PORT=(388:4,530:8)
;
```

```
;
;       Example 4: To configure the Proteon Token Ring card using
;       IRQ A, DMA channel 7, I/O Ports A20-A3F, STP media and
;       speed of 16 Mbits remove the comment characters from the
;       following commands.
;
;       [IRA0]
;        NAME=IR
;        NODE=4
;        DRIVER=SYS$IRDRIVER
;        IRQ=A
;        DMA=7
;        PORT=(A20:20)
;        USER_PARAM="STP,16"
;
;
;   System Board Resources for AlphaStation 200 and 400 Series Systems
;   ------------------------------------------------------------------
;
;    COM1:
;        IRQ=4, I/O port=3F8:8
;
;    COM2:
;        IRQ=3, I/O port=2F8:8
;
;    LPT1:
;        IRQ=7, I/O Port=378:8
;
;    FDC:
;        IRQ=6, I/O port=3F0:8, DMA chan=2
;
;    Mouse:
;        IRQ=C, I/O port=60,64
;
;    Keyboard:
;        IRQ=1, I/O port=60,64
;
;    TOY Clock:
;        I/O port=70,71
;
;    PCI NCR810 SCSI:
;        IRQ=B, I/O port=26,27
;
;
;    Sound: (This is a system board device on AlphaStation 200 Series only)
;        IRQ=9, I/O Ports=388:4,530:8, DMA chan=(0,1)
;
;    Timer/Counter:
;        IRQ=0
;
;    INTEL SIO PCI/ISA Bridge interrupt logic:
;        IRQ=2 (Cascade IRQ for the dual 8259 on the bridge chip)
;        IRQ=D (used for DMA Buffer Chaining, unconnnected on these systems)
;        IRQ=8 (used for interval timer, unconnected on these systems)
;        DMA=4 (used by the DMA controller as the cascade line)
;
```

B

Remedial Kits Included in OpenVMS Version 6.2

This appendix lists remedial kits that are included in OpenVMS Version 6.2. Appendix C describes other remedial kits that are *not* included in Version 6.2, but that will be available in a future release.

Digital updates existing kits and creates new kits as necessary. Contact your Digital support representative for the latest information about new remedial kits.

The following sections list the remedial kits included in Version 6.2 of the OpenVMS VAX and OpenVMS Alpha operating systems. For some older kits that were originally assigned a CSCPAT number, that number is shown in parentheses.

Kit names are constructed from the following information in this order:

- Platform name: VAX or AXP (for Alpha)
- Facility name, abbreviated to 4 characters if necessary
- Number of the kit for this facility for this version
- Version number

For example, VAXBACK02_061 is the second remedial kit that has been created for the Backup utility in Version 6.1 of OpenVMS VAX.

B.1 Remedial Kits Included in OpenVMS VAX Version 6.2

VAX

The following remedial kits are included in Version 6.2 of the OpenVMS VAX operating system:

VAXACRT01_061
VAXACRT02_061
VAXAUDI01_061
VAXAUDI03_061
VAXBACK02_060
VAXBACK02_061
VAXBACK03_U2055
VAXBACK4_U2055
VAXBOOT01_061
VAXCDU01_060
VAXCDU01_U2055
VAXCLIU01_061
VAXCLIU02_U2055
VAXCMAR01_061
VAXCOPY01_061
VAXCXXL01_061
VAXDDTM01_U2055

VAXDDTM02_061
VAXDEBU02_061
VAXDECW01_060
VAXDECW02_060
VAXDECW11_U2055
VAXDUP01_060
VAXDWXT01_061
VAXF11C02_061
VAXF11X03_061
VAXINIT03_061
VAXLAT02_U2055
VAXLMF01_061
VAXLOAD01_061
VAXLOGI04_061
VAXMAIL02_060
VAXMAIL03_061
VAXMAIL05_061
VAXMAIL06_U2055
VAXMONT01_061
VAXMOTF03_011
VAXMOUN01_061
VAXNETA01_061
VAXPHV_02_061
VAXPHV_03_060
VAXPHV_09_U2055
VAXQMAN03_061
VAXRMS01_061
VAXSCSI02_061
VAXSCSI03_061
VAXSHAD02_060
VAXSORT01_061
VAXSYS01_061
VAXSYS02_061
VAXSYS04_061
VAXSYS09_U2055
VAXSYS10_U2055
VAXSYS12_U2055
VAXSYS12_061
VAXSYS13_061
VAXSYSL01_061
VAXSYSL01_2H455
VAXSYSL09_U1055
VAXSYSL11_U1055
VAXSYSL17_U2055
VAXTTDR09_U2055
VAXVERI02_061 ♦

B.2 Remedial Kits Included in OpenVMS Alpha Version 6.2

Alpha

The following remedial kits are included in Version 6.2 of the OpenVMS Alpha operating system:

AXPACRT03_061
AXPACRT04_061
AXPAMAC02_U1061
AXPBOOT03_U1061
AXPBOOT04_U1061
AXPCMAR01_U1061
AXPCMAR01_061
AXPCOB01_U1061
AXPCOPY01_061
AXPCXXL01_061
AXPDEBU02_061
AXPDPLI01_061
AXPDRIV07_061
AXPDRIV08_061
AXPDWXT01_015
AXPDWXT01_061
AXPF11C02_061
AXPINIT01_061
AXPLAN03_061
AXPLAN04_061
AXPLAT01_061
AXPLAT03_061
AXPLINK02_061
AXPLOGI01_061
AXPMTAA01_061
AXPOPDR01_015
AXPOPDR04_061
AXPQMAN01_015
AXPQMAN03_061
AXPSCSI01_061
AXPSHAD04_061
AXPSORT01_061
AXPSYS07_061
AXPSYS10_061
AXPSYS11_061
AXPTTDR01_061
AXPTTDR02_015
AXPVERI02_061 ♦

C

Remedial Kits Not Included in OpenVMS Version 6.2

This appendix describes remedial kits that are *not* included in OpenVMS Version 6.2. Appendix B lists all remedial kits that *are* included in Version 6.2 of the OpenVMS operating system.

Digital updates existing kits and creates new kits as necessary. Contact your Digital support representative for the latest information about new remedial kits.

Kit names are constructed from the following information in this order:

- Platform name: VAX or AXP (for Alpha)
- Facility name, abbreviated to 4 characters if necessary
- Number of the kit for this facility for this version
- Version number

For example, AXPTRAC01_061 is the first remedial kit to be created for the Trace facility in Version 6.1 of OpenVMS Alpha.

C.1 Remedial Kits Not Included in OpenVMS VAX Version 6.2

VAX

The following sections describe remedial kits that are *not* included in Version 6.2 of the OpenVMS VAX operating system. These kits will likely be included in the next release of the operating system.

C.1.1 Local Area VMScluster Kit (VAXLAVC01_061)

When this remedial kit is applied, nodes in a Local Area VMScluster environment no longer invoke a CLUEXIT bugcheck in mixed Ethernet/FDDI environments where hello messages are being produced (that is, in outdated bridge firmware). ♦

C.2 Remedial Kits Not Included in OpenVMS Alpha Version 6.2

Alpha

The following sections describe remedial kits that are *not* included in Version 6.2 of the OpenVMS Alpha operating system. These kits will likely be included in the next release of the operating system.

C.2.1 Trace Facility (AXPTRAC01_061)

This kit corrects a problem where traceback information reported after a program failure was either incomplete or incorrect. ♦

D

Layered Product Support for OpenVMS Version 6.2

This appendix provides preliminary information about layered products that are compatible with OpenVMS Version 6.2.

Tables in this appendix list layered products that support or will soon support OpenVMS Version 6.2. The maximum version of the operating system on which each layered product has been qualified is shown in the Operating System Support (Qualified) column. Most layered products that supported OpenVMS Version 6.1 will also support OpenVMS Version 6.2.

The tables also indicate the DECnet environments supported by each layered product. Support categories are abbreviated in the tables as Ph. IV (DECnet Phase IV), Syn (DECnet/OSI synonyms), and Full (DECnet/OSI full names).

Most layered products that worked in an OpenVMS Version 6.1 environment running DECnet for OpenVMS (DECnet Phase IV) will also work in an OpenVMS Version 6.2 environment running DECnet for OpenVMS. Many of these layered products will also work in a DECnet/OSI synonyms environment.

Regardless of the DECnet environment, the following layered products require these new versions for installation on OpenVMS Version 6.2:

- DEC X.500 Directory Services, Version 2.0A (Administration Facility)
- DEC X.500 Directory Services, Version 2.0A (Starter Kit)
- DEC X.500 Directory Services, Version 2.0A (Server)
- DECdfs for OpenVMS Systems, Version 2.1
- POLYCENTER Capacity Planner for OpenVMS, Version 3.0A
- StorageWorks RAID Software for OpenVMS, Version 2.2

Table D–1 lists layered products compatible with OpenVMS VAX Version 6.2 that will be available at the same time as or shortly following the release of OpenVMS VAX Version 6.2.

Table D–2 lists layered products compatible with OpenVMS Alpha Version 6.2 that will be available at the same time as or shortly following the release of OpenVMS Alpha Version 6.2.

Table D–3 lists layered products compatible with OpenVMS VAX Version 6.2 that are planned for release after May 1995.

Table D–4 lists layered products compatible with OpenVMS Alpha Version 6.2 that are planned for release after May 1995.

Information in these tables is continually evolving and is subject to change. As additional results from ongoing qualification tests become available, updated information will be included on the OpenVMS Software Product Library kits (CD–ROM consolidations). Digital now regularly compiles an OpenVMS Layered Product Public Rollout Schedule for both VAX and Alpha systems. For the most current information about OpenVMS layered products, please contact your Digital support representative.

Table D–1 Digital Layered Products Compatible with OpenVMS VAX Version 6.2

Product and Version	Operating System Support	DECnet Support		
	(Qualified)	Ph. IV	Syn	Full
ACMS Desktop, V2.0	6.1	Yes	Yes	No
ACMS Desktop/Japanese, V2.0	6.1	Yes	Yes	No
ACMS for OpenVMS (Development), V4.0	6.1	Yes	N/A	N/A
ACMS for OpenVMS (Remote), V4.0	6.1	Yes	N/A	N/A
ACMS for OpenVMS (Run-Time), V4.0	6.1	Yes	N/A	N/A
ALL–IN–1 DESKtop Server for VMS, V1.2	6.1	Yes	Yes	No
ALL–IN–1 Integrated Office System Server for VMS (MUPA), V3.0B	6.1	Yes	Yes	No
ALL–IN–1 Integrated Office System Server for VMS (MUPA), V3.1	6.1	Yes	Yes	No
ALL–IN–1 IOS TeamLinks Connection V2.1 for OpenVMS VAX, V3.0C	6.1	Yes	Yes	No
ALL–IN–1 IOS TeamLinks Connection V2.1 for OpenVMS VAX, V3.1	6.1	Yes	Yes	No
ALL–IN–1 Office Server Options for OpenVMS AXP and OpenVMS VAX, V3.0B	6.1	Yes	Yes	No
ALL–IN–1 Office Server Options for OpenVMS AXP and OpenVMS VAX, V3.1	6.1	Yes	Yes	No
ALL–IN–1 Personal Assistant for OpenVMS, V1.1	6.1	Yes	Yes	No
ALL–IN–1/ Français Integrated Office System Server for OpenVMS, V3.0B	6.1	Yes	Yes	No
ALL–IN–1/ Français Integrated Office System Server for OpenVMS, V3.1	6.1	Yes	Yes	No
ALL–IN–1/British Integrated Office System Server for OpenVMS, V3.0B	6.1	Yes	Yes	No
ALL–IN–1/Dansk Integrated Office System Server for OpenVMS, V3.0B	6.1	Yes	Yes	No
ALL–IN–1/Dansk Integrated Office System Server for OpenVMS, V3.1	6.1	Yes	Yes	No
ALL–IN–1/Dansk Office Server Options for OpenVMS AXP and VAX, V3.0B	6.1	Yes	Yes	No
ALL–IN–1/Dansk Office Server Options for OpenVMS AXP and VAX, V3.1	6.1	Yes	Yes	No
ALL–IN–1/Deutsch Integrated Office System Server for OpenVMS, V3.0B	6.1	Yes	Yes	No
ALL–IN–1/Deutsch Integrated Office System Server for OpenVMS, V3.1	6.1	Yes	Yes	No
ALL–IN–1/Deutsch Office Server Options for OpenVMS AXP and VAX, V3.0B	6.1	Yes	Yes	No
ALL–IN–1/Deutsch Office Server Options for OpenVMS AXP and VAX, V3.1	6.1	Yes	Yes	No
ALL–IN–1/Español Integrated Office System Server for OpenVMS, V3.0B	6.1	Yes	Yes	No
ALL–IN–1/Español Integrated Office System Server for OpenVMS, V3.1	6.1	Yes	Yes	No
ALL–IN–1/Español Office Server Options for OpenVMS AXP and VAX, V3.0B	6.1	Yes	Yes	No
ALL–IN–1/Español Office Server Options for OpenVMS AXP and VAX, V3.1	6.1	Yes	Yes	No
ALL–IN–1/Français Office Server Options for OpenVMS AXP and VAX, V3.0B	6.1	Yes	Yes	No

(continued on next page)

Table D–1 (Cont.) Digital Layered Products Compatible with OpenVMS VAX Version 6.2

Product and Version	Operating System Support	DECnet Support		
	(Qualified)	Ph. IV	Syn	Full
ALL–IN–1/Français Office Server Options for OpenVMS AXP and VAX, V3.1	6.1	Yes	Yes	No
ALL–IN–1/Hebrew Integrated Office System Server for OpenVMS, V3.0	6.1	Yes	Yes	No
ALL–IN–1/Italiano Integrated Office System Server for OpenVMS, V3.0B	6.1	Yes	Yes	No
ALL–IN–1/Italiano Integrated Office System Server for OpenVMS, V3.1	6.1	Yes	Yes	No
ALL–IN–1/Italiano Office Server Options for OpenVMS AXP and VAX, V3.0B	6.1	Yes	Yes	No
ALL–IN–1/Italiano Office Server Options for OpenVMS AXP and VAX, V3.1	6.1	Yes	Yes	No
ALL–IN–1/Japanese (English Secondary), V3.0B	6.1	Yes	Yes	No
ALL–IN–1/Japanese Integrated Office System Server for OpenVMS, V3.0B	6.1	Yes	Yes	No
ALL–IN–1/Nederlands Integrated Office System Server for OpenVMS, V3.0B	6.1	Yes	Yes	No
ALL–IN–1/Nederlands Integrated Office System Server for OpenVMS, V3.1	6.1	Yes	Yes	No
ALL–IN–1/Nederlands Office Server Options for OpenVMS AXP and VAX, V3.0B	6.1	Yes	Yes	No
ALL–IN–1/Nederlands Office Server Options for OpenVMS AXP and VAX, V3.1	6.1	Yes	Yes	No
ALL–IN–1/Norsk Integrated Office System Server for OpenVMS, V3.0B	6.1	Yes	Yes	No
ALL–IN–1/Norsk Integrated Office System Server for OpenVMS, V3.1	6.1	Yes	Yes	No
ALL–IN–1/Norsk Office Server Options for OpenVMS AXP and VAX, V3.0B	6.1	Yes	Yes	No
ALL–IN–1/Norsk Office Server Options for OpenVMS AXP and VAX, V3.1	6.1	Yes	Yes	No
ALL–IN–1/Portugûes Integrated Office System Server for OpenVMS, V3.0B	6.1	Yes	Yes	No
ALL–IN–1/Portugûes Integrated Office System Server for OpenVMS, V3.1	6.1	Yes	Yes	No
ALL–IN–1/Portugûes Office Server Options for OpenVMS AXP and VAX, V3.0B	6.1	Yes	Yes	No
ALL–IN–1/Portugûes Office Server Options for OpenVMS AXP and VAX, V3.1	6.1	Yes	Yes	No
ALL–IN–1/Suomi Integrated Office System Server for OpenVMS, V3.0B	6.1	Yes	Yes	No
ALL–IN–1/Suomi Integrated Office System Server for OpenVMS, V3.1	6.1	Yes	Yes	No
ALL–IN–1/Suomi Office Server Options for OpenVMS AXP and VAX, V3.0B	6.1	Yes	Yes	No
ALL–IN–1/Suomi Office Server Options for OpenVMS AXP and VAX, V3.1	6.1	Yes	Yes	No
ALL–IN–1/Svenska Integrated Office System Server for OpenVMS, V3.0B	6.1	Yes	Yes	No
ALL–IN–1/Svenska Integrated Office System Server for OpenVMS, V3.1	6.1	Yes	Yes	No
ALL–IN–1/Svenska Office Server Options for OpenVMS AXP and VAX, V3.0B	6.1	Yes	Yes	No
ALL–IN–1/Svenska Office Server Options for OpenVMS AXP and VAX, V3.1	6.1	Yes	Yes	No
BASEstar Classic for OpenVMS VAX (Development), V3.3	6.1	Yes	Yes	No
BASEstar Classic for OpenVMS VAX (Development), V3.4	6.1	Yes	Yes	No
BASEstar Classic for OpenVMS VAX (Run-Time), V3.3	6.1	Yes	Yes	No
BASEstar Classic for OpenVMS VAX (Run-Time), V3.4	6.1	Yes	Yes	No
BASEstar Open Client for OpenVMS VAX, V2.0A	6.1	N/A	N/A	N/A
BASEstar Open Server for OpenVMS VAX (Development), V2.0A	6.1	N/A	N/A	N/A
BASEstar Open Server for OpenVMS VAX (Run-Time), V2.0A	6.1	N/A	N/A	N/A
CDD/Administrator for OpenVMS Systems, V1.2	6.1	Yes	No	No

(continued on next page)

Table D–1 (Cont.) Digital Layered Products Compatible with OpenVMS VAX Version 6.2

Product and Version	Operating System Support	DECnet Support		
	(Qualified)	Ph. IV	Syn	Full
CDD/Administrator for OpenVMS Systems, V1.3	6.1	Yes	Yes	No
CDD/Repository/Japanese for OpenVMS VAX, V5.1	6.1	Yes	No	No
Consolidated DEChub Firmware Kit, V3.1	6.1	Yes	No	No
DEC @a Glance for VMS (Client Run-Time Option), V1.0	6.1	Yes		
DEC @a Glance for VMS (Development Option), V1.0	6.1	Yes		
DEC @a Glance for VMS (Server Run-Time Option), V1.0	6.1	Yes		
DEC Ada for OpenVMS VAX Systems, V3.0A	6.1	Yes	Yes	No
DEC Ada for OpenVMS VAX Systems, V3.2	6.2	Yes	Yes	No
DEC Ada Professional Development Option for OpenVMS VAX Systems, V3.0A	6.1	Yes	Yes	No
DEC Ada Professional Development Option for OpenVMS VAX Systems, V3.2	6.2	Yes	Yes	No
DEC C for OpenVMS VAX Systems, V4.0	6.2	N/A	N/A	N/A
DEC C for OpenVMS VAX Systems, V5.0	6.2	N/A	N/A	N/A
DEC C++ for OpenVMS VAX (Compiler), V1.2	6.1	N/A	N/A	N/A
DEC C++ for OpenVMS VAX (Compiler), V5.0	6.2	N/A	N/A	N/A
DEC C++ for OpenVMS VAX (VAX Debugger), V1.2	6.1	N/A	N/A	N/A
DEC C++ for OpenVMS VAX (VAX Debugger), V5.0	6.2	N/A	N/A	N/A
DEC C/C++ Run-Time Components for OpenVMS VAX, V1.2	6.1	N/A	N/A	N/A
DEC C/C++ Run-Time Components for OpenVMS VAX, V5.0	6.2	N/A	N/A	N/A
DEC C/Japanese for OpenVMS VAX, V4.0	6.1	N/A	N/A	N/A
DEC Code Management System (CMS) for OpenVMS VAX Systems, V3.6	6.1	Yes	No	No
DEC Code Management System (CMS) for OpenVMS VAX Systems, V3.7	6.2	Yes	No	No
DEC COMMserver for OpenVMS, V2.0	6.1	Yes	No	No
DEC COMMserver for OpenVMS, V3.0	6.1	Yes	No	No
DEC Computer Integrated Telephony Application Interface for VMS, V3.0A	6.1	Yes	No	No
DEC Data Distributor for OpenVMS VAX, V6.1	6.1	Yes		
DEC DATATRIEVE for OpenVMS VAX, V7.0	6.1	Yes	Yes	Yes
DEC DATATRIEVE/Hangul for OpenVMS VAX, V6.1	6.1	Yes	Yes	No
DEC DATATRIEVE/Japanese for OpenVMS VAX, V6.1	6.1	Yes	Yes	No
DEC DB Integrator Gateway for DBMS for OpenVMS VAX, V1.0	6.1	Yes	Yes	No
DEC DB Integrator Gateway for DSM, V1.0B	6.1	Yes		
DEC DB Integrator Gateway for EDA/SQL, V3.1	6.1	Yes		
DEC DB Integrator V1.0B for OpenVMS VAX (Development), V4.0	6.1	Yes		
DEC DB Integrator V1.0B for OpenVMS VAX (Interactive), V4.0	6.1	Yes		
DEC DB Integrator V1.0B for OpenVMS VAX (Run-Time), V4.0	6.1	Yes		
DEC DBA Workcenter for OpenVMS VAX, V1.0	6.1	Yes		
DEC Distributed Queuing Service for OpenVMS, V1.3	6.2	Yes	Yes	Yes

(continued on next page)

Table D–1 (Cont.) Digital Layered Products Compatible with OpenVMS VAX Version 6.2

Product and Version	Operating System Support	DECnet Support		
	(Qualified)	Ph. IV	Syn	Full
DEC Fortran for OpenVMS Systems, V6.3	6.2	Yes	Yes	Yes
DEC GKS for OpenVMS, V6.0	6.1	Yes	Yes	Yes
DEC GKS for OpenVMS, V6.1	6.1	Yes	Yes	Yes
DEC GKS for OpenVMS (Run-Time), V6.0	6.1	Yes	Yes	Yes
DEC GKS for OpenVMS (Run-Time), V6.1	6.1	Yes	Yes	Yes
DEC Language-Sensitive Editor/Source Code Analyzer for OpenVMS VAX, V4.2	6.1	Yes	No	No
DEC Language-Sensitive Editor/Source Code Analyzer for OpenVMS VAX, V4.3	6.2	Yes	No	No
DEC MAILworks/Japanese for OpenVMS, V1.2	6.1	Yes	Yes	No
DEC Module Management System (MMS) for OpenVMS VAX Systems, V2.7	6.1	Yes	No	No
DEC Module Management System (MMS) for OpenVMS VAX Systems, V3.0	6.2	Yes	No	No
DEC OPS5 for OpenVMS, V4.0A	6.1	Yes	Yes	Yes
DEC OPS5/Japanese for OpenVMS, V4.0A	6.1	Yes	Yes	Yes
DEC Optical Storage Desktop Software (OSDS) for OpenVMS, V3.2	6.1	N/A	N/A	N/A
DEC Optical Storage Desktop Software (OSDS) for OpenVMS, V3.3	6.1	N/A	N/A	N/A
DEC Optical Storage Management Software (OSMS) for OpenVMS, V3.2	6.1	N/A	N/A	N/A
DEC Optical Storage Management Software (OSMS) for OpenVMS, V3.3	6.1	N/A	N/A	N/A
DEC Pascal for OpenVMS, V5.1	6.1	Yes	Yes	Yes
DEC Performance and Coverage Analyzer for OpenVMS VAX Systems, V4.2	6.1	Yes	No	No
DEC Performance and Coverage Analyzer for OpenVMS VAX Systems, V4.3	6.2	Yes	No	No
DEC PHIGS for OpenVMS VAX, V3.0	6.1	Yes	Yes	Yes
DEC PHIGS for OpenVMS VAX, V3.1	6.2	Yes	N/A	N/A
DEC PHIGS for OpenVMS VAX (Run-Time), V3.0	6.1	Yes	Yes	Yes
DEC PHIGS for OpenVMS VAX (Run-Time), V3.1	6.2	Yes	N/A	N/A
DEC PHIGS/Japanese for VMS (Run-Time), V3.0A	6.1	Yes	Yes	Yes
DEC PHIGS/Japanese for VMS VAX, V3.0A	6.1	Yes	Yes	Yes
DEC Rdb/Hangul for OpenVMS/Hangul VAX (Multi-Version), V6.0	6.1	Yes	Yes	No
DEC Rdb/Hangul for OpenVMS/Hangul VAX (Standard), V6.0	6.1	Yes	Yes	No
DEC Rdb/Hanyu for OpenVMS/Hanyu VAX (Multi-Version), V6.0	6.1	Yes	Yes	No
DEC Rdb/Hanyu for OpenVMS/Hanyu VAX (Standard), V6.0	6.1	Yes	Yes	No
DEC Rdb/Hanzi for OpenVMS/Hanzi VAX (Multi-Version), V6.0	6.1	Yes	Yes	No
DEC Rdb/Hanzi for OpenVMS/Hanzi VAX (Standard), V6.0	6.1	Yes	Yes	No
DEC Rdb/Japanese for OpenVMS VAX (Interactive), V6.0	6.1	Yes	Yes	No
DEC Rdb/Japanese for OpenVMS VAX (Run-Time), V6.0	6.1	Yes	Yes	No
DEC Rdb/Japanese for OpenVMS/Japanese VAX (Full Development), V6.0	6.1	Yes	Yes	No
DEC SecurityGate for OpenVMS, V1.1A	6.1	Yes	No	No
DEC SERdb for Security-Enhanced VMS, V6.0	6.1	Yes	Yes	No

(continued on next page)

Table D–1 (Cont.) Digital Layered Products Compatible with OpenVMS VAX Version 6.2

Product and Version	Operating System Support	DECnet Support		
	(Qualified)	Ph. IV	Syn	Full
DEC SNA 3270 Application Services (Development), V1.1	6.1	Yes	Yes	No
DEC SNA 3270 Application Services (Run-Time), V1.1	6.1	Yes	Yes	No
DEC SNA Domain Gateway, V2.0	6.1	Yes	Yes	No
DEC SoftPC for OpenVMS, V4.0	6.1	Yes	No	No
DEC TCP/IP Services for OpenVMS, V3.3	6.1	N/A	N/A	N/A
DEC TCP/IP/Japanese Services for OpenVMS/Japanese, V3.2	6.1	N/A	N/A	N/A
DEC Test Manager for OpenVMS VAX Systems, V3.5	6.1	Yes	No	No
DEC Test Manager for OpenVMS VAX Systems, V3.6	6.2	Yes	No	No
DEC VOICEmail for OpenVMS VAX, V1.3A	6.1	Yes	No	No
DEC VTX Version OpenVMS VAX, V6.2	6.1	Yes	No	No
DEC X.500 Directory Administration Facility for OpenVMS VAX, V2.0	6.1	N/A	Yes	Yes
DEC X.500 Directory Administration Facility for OpenVMS VAX, V2.0A	6.2	N/A	Yes	Yes
DEC X.500 Directory Server for OpenVMS VAX, V2.0	6.1	N/A	Yes	Yes
DEC X.500 Directory Server for OpenVMS VAX, V2.0A	6.2	N/A	Yes	Yes
DEC X.500 Directory Service Starter Kit for OpenVMS VAX, V2.0	6.1	N/A	Yes	Yes
DEC X.500 Directory Service Starter Kit for OpenVMS VAX, V2.0A	6.2	N/A	Yes	Yes
DEC/EDI for OpenVMS VAX, V2.0	6.1	Yes	Yes	No
DEC/EDI for OpenVMS VAX, V2.1	6.2	Yes	Yes	No
DECADMIRE, V2.1A	6.1	N/A	N/A	N/A
DECADMIRE w/Hebrew Support for ACMS/COBOL/DECforms for OpenVMS, V1.2	6.1	N/A	N/A	N/A
DECADMIRE w/Hebrew Support for COBOL/DECforms for OpenVMS, V1.2	6.1	N/A	N/A	N/A
DECADMIRE/Japanese for OpenVMS, V2.0	6.1	N/A	N/A	N/A
DECdfs for OpenVMS Systems, V2.0	6.2	Yes	Yes	No
DECdfs for OpenVMS Systems, V2.1	6.2	Yes	Yes	No
DECforms for OpenVMS VAX (Development), V2.0	6.1	N/A	N/A	N/A
DECforms for OpenVMS VAX (Development), V2.1	6.1	N/A	N/A	N/A
DECforms for OpenVMS VAX (Run-Time), V2.0	6.1	N/A	N/A	N/A
DECforms for OpenVMS VAX (Run-Time), V2.1	6.1	N/A	N/A	N/A
DECforms/Hangul for OpenVMS VAX (Development), V2.0	6.1	N/A	N/A	N/A
DECforms/Hangul for OpenVMS VAX (Run-Time), V2.0	6.1	N/A	N/A	N/A
DECforms/Hanyu for OpenVMS VAX (Development), V2.0	6.1	N/A	N/A	N/A
DECforms/Hanyu for OpenVMS VAX (Run-Time), V2.0	6.1	N/A	N/A	N/A
DECforms/Hanzi for OpenVMS VAX (Development), V2.0	6.1	N/A	N/A	N/A
DECforms/Hanzi for OpenVMS VAX (Run-Time), V2.0	6.1	N/A	N/A	N/A
DECforms/Japanese, V2.0	6.1	N/A	N/A	N/A
DECforms/Japanese (Run-Time), V2.0	6.1	N/A	N/A	N/A

(continued on next page)

Table D–1 (Cont.) Digital Layered Products Compatible with OpenVMS VAX Version 6.2

Product and Version	Operating System Support	DECnet Support		
	(Qualified)	Ph. IV	Syn	Full
DECimage Application Services for VMS, V3.2	6.1	N/A	N/A	N/A
DECimage Express, V2.2A	6.1	Yes		
DECimage Express, V2.3	6.1	Yes		
DECintact, V2.1	6.1	Yes	No	No
DECintact (Remote), V2.1	6.1	Yes	No	No
DECintact (Run-Time), V2.1	6.1	Yes	No	No
DECmessageQ for OpenVMS VAX, V2.1B	6.1	Yes	No	No
DECmessageQ for OpenVMS VAX, V3.0	6.2	Yes	Yes	No
DECnet SNA 3270 Data Stream Programming Interface for OpenVMS, V1.4A	6.1	Yes	Yes	No
DECnet SNA 3270 Terminal Emulator for OpenVMS, V1.5A	6.1	Yes	Yes	No
DECnet SNA 3270 Terminal Emulator/Japanese for VMS/Japanese, V1.5A	6.1	Yes	No	No
DECnet SNA APPC/LU6.2 Programming Interface for OpenVMS, V2.2A	6.1	Yes	Yes	No
DECnet SNA Application Programming Interface for OpenVMS, V2.3A	6.1	Yes	Yes	No
DECnet SNA Data Transfer Facility (Server), V3.1A	6.1	Yes	Yes	No
DECnet SNA Data Transfer Facility (Utilities), V3.1A	6.1	Yes	Yes	No
DECnet SNA Gateway for Channel Transport, V2.1A	6.1	Yes	Yes	No
DECnet SNA Gateway for Synchronous Transport, V1.2A	6.1	Yes	Yes	No
DECnet SNA Remote Job Entry for OpenVMS, V1.4A	6.1	Yes	Yes	No
DECnet SNA Remote Job Entry/Japanese for OpenVMS, V1.4A	6.1	Yes	Yes	No
DECnet SNA VMS Printer Emulator, V1.2A	6.1	Yes	Yes	No
DECnet SNA VMS Printer Emulator/Japanese, V1.2A	6.1	Yes	No	No
DECnet/OSI Version for OpenVMS VAX, V6.1	6.1	N/A	Yes	Yes
DECnet/OSI Version for OpenVMS VAX, V6.2	6.2	N/A	Yes	Yes
DEComni for OpenVMS VAX, V2.1	6.1	N/A	N/A	N/A
DECosap for OpenVMS VAX, V2.0B	6.1	N/A	N/A	N/A
DECprint Supervisor (DCPS) for OpenVMS, V1.1A	6.1	Yes	No	No
DECprint Supervisor (DCPS) for OpenVMS, V1.2	6.1	Yes	No	No
DECprint Supervisor/Japanese for OpenVMS, V1.1	6.1	Yes	No	No
DECram for OpenVMS, V2.1	6.2	N/A	N/A	N/A
DECset for OpenVMS VAX Systems, V11.2	6.1	Yes	No	No
DECset for OpenVMS VAX Systems, V12.0	6.2	Yes	No	No
DECvoice Software, V2.3	6.1	N/A	N/A	N/A
DECvoiceBuilder Software for OpenVMS (Administration and Run-Time), V1.1A	6.1	N/A	N/A	N/A
DECvoiceBuilder Software for OpenVMS (Edit and Record Studio), V1.1A	6.1	N/A	N/A	N/A
DECwatchdog/Autopilot, V2.1	6.1	No	N/A	N/A
DECwindows DECnet SNA 3270 Terminal Emulator for OpenVMS, V2.0A	6.1	Yes	Yes	No

(continued on next page)

Table D–1 (Cont.) Digital Layered Products Compatible with OpenVMS VAX Version 6.2

Product and Version	Operating System Support	DECnet Support		
	(Qualified)	Ph. IV	Syn	Full
DECwindows DECnet SNA 3270 Terminal Emulator/Japanese for OpenVMS, V2.0A	6.1	Yes	No	No
DECwindows Motif for OpenVMS, V1.2-3	6.2	Yes	Yes	Yes
DECwindows Motif OpenVMS User Interface/Cesky, V1.2	6.1	Yes	Yes	Yes
DECwindows Motif OpenVMS User Interface/Deutsch, V1.2	6.1	Yes	Yes	Yes
DECwindows Motif OpenVMS User Interface/Español, V1.2	6.1	Yes	Yes	Yes
DECwindows Motif OpenVMS User Interface/Français, V1.2	6.1	Yes	Yes	Yes
DECwindows Motif OpenVMS User Interface/Italiano, V1.2	6.1	Yes	Yes	Yes
DECwindows Motif OpenVMS User Interface/Magyar, V1.2	6.1	Yes	Yes	Yes
DECwindows Motif OpenVMS User Interface/Polski, V1.2	6.1	Yes	Yes	Yes
DECwindows Motif OpenVMS User Interface/Russkij, V1.2	6.1	Yes	Yes	Yes
DECwindows Motif OpenVMS User Interface/Slovenski, V1.2	6.1	Yes	Yes	Yes
DECwindows Motif OpenVMS User Interface/Svenska, V1.2	6.1	Yes	Yes	Yes
DECwindows Motif/Hangul for OpenVMS, V1.2	6.1	Yes	Yes	Yes
DECwindows Motif/Hanyu for OpenVMS, V1.2	6.1	Yes	Yes	Yes
DECwindows Motif/Hanzi for OpenVMS, V1.2	6.1	Yes	Yes	Yes
DECwindows Motif/Japanese for OpenVMS, V1.2	6.1	Yes	Yes	No
DECwindows Motif/Thai for OpenVMS, V1.2	6.1	Yes	Yes	Yes
DECwindows/Hebrew for OpenVMS, V1.2	6.1	Yes	Yes	No
DECwrite for OpenVMS VAX, V3.0	6.1	N/A	N/A	N/A
DECwrite/British for OpenVMS VAX, V3.0	6.1	N/A	N/A	N/A
DECwrite/Deutsch for VMS, V3.0	6.1	N/A	N/A	N/A
DECwrite/Français for VMS, V3.0	6.1	N/A	N/A	N/A
Digital Extended Math Library for OpenVMS (Development), V2.0A	6.1	N/A	N/A	N/A
Digital Extended Math Library for OpenVMS (Run-Time), V2.0A	6.1	N/A	N/A	N/A
Distributed Computing Environment (DCE) for OpenVMS (Development), V1.3	6.2			
Distributed Computing Environment (DCE) for OpenVMS (Run-Time), V1.3	6.2			
DSM Version for OpenVMS VAX Systems, V6.3D	6.1	Yes	No	No
DSM Version for OpenVMS VAX Systems, V6.4	6.1	Yes	No	No
EDCS II (Client), V2.2	6.1	Yes	No	No
EDCS II (Server), V2.2	6.1	Yes	No	No
IEX-VMS-Driver, V4.4	6.1	N/A	N/A	N/A
InfoServer Software, V3.2	6.1	Yes		
KAP for DEC C for OpenVMS VAX, V1.1	6.1	N/A	N/A	N/A
KAP for DEC Fortran for OpenVMS VAX, V1.1	6.1	N/A	N/A	N/A
MAILbus 400 Application Program Interface for OpenVMS, V1.4	6.1	Yes	N/A	N/A

(continued on next page)

Table D–1 (Cont.) Digital Layered Products Compatible with OpenVMS VAX Version 6.2

Product and Version	Operating System Support	DECnet Support		
	(Qualified)	Ph. IV	Syn	Full
MAILbus 400 Message Transfer Agent for OpenVMS, V1.4	6.1	Yes	N/A	N/A
MAILbus 400/Message Router Gateway, V1.1	6.1	No	Yes	No
MAILWorks for OpenVMS, V1.3A	6.1	Yes	Yes	No
MEP VT for DEC SNA 3270 Application Services, V3.0	6.1	Yes	Yes	No
MicroVAX/DRQ3B Device Driver, V1.4	6.1	N/A	N/A	N/A
MUXserver 320/380/90 Remote Terminal Server, V2.0	6.2	Yes	No	No
NAS Base Server 200 for OpenVMS VAX Systems, V6.0	6.2	N/A	N/A	N/A
NAS Client 150 for OpenVMS VAX Workstations, V6.0	6.2	N/A	N/A	N/A
NAS Client 250 for OpenVMS VAX Workstations, V6.0	6.2	N/A	N/A	N/A
NAS Production Server 400 for OpenVMS VAX Systems, V6.0	6.2	N/A	N/A	N/A
NAS Server 300 for OpenVMS VAX Systems, V6.0	6.2	N/A	N/A	N/A
ObjectBroker for OpenVMS (Development), V2.5A	6.2	Yes	No	No
ObjectBroker for OpenVMS (Run-Time), V2.5A	6.2	Yes	No	No
OpenVMS SNA, V2.2	6.1	Yes	Yes	No
OpenVMS VAX Operating System, V6.1	6.1	Yes	Yes	Yes
OpenVMS VAX Operating System, V6.2	6.2	Yes	Yes	Yes
OpenVMS/Hangul VAX Operating System, V6.1	6.1			
OpenVMS/Hanyu VAX Operating System, V6.1	6.1			
OpenVMS/Hanzi VAX Operating System, V6.1	6.1			
OpenVMS/Japanese VAX Operating System, V6.1	6.1			
OpenVMS/Thai VAX Operating System, V6.1	6.1			
Oracle CDD/Repository for OpenVMS Systems, V5.4	6.1	Yes	Yes	No
Oracle CODASYL DBMS (Development), V6.0A	6.2	Yes	Yes	No
Oracle CODASYL DBMS (Development), V6.1	6.2	Yes	Yes	No
Oracle CODASYL DBMS (Run-Time), V6.0A	6.2	Yes	Yes	No
Oracle CODASYL DBMS (Run-Time), V6.1	6.2	Yes	Yes	No
Oracle Expert, V2.1A	6.1	N/A	N/A	N/A
Oracle Graphical Schema Editor (GSE) for Rdb, OpenVMS, V2.1A	6.1	N/A	N/A	N/A
Oracle InstantSQL for Rdb, V2.1A	6.1	N/A	N/A	N/A
Oracle RALLY, V4.0A	6.1	Yes	Yes	Yes
Oracle RALLY, V4.1	6.1	Yes	Yes	Yes
Oracle RALLY (Development), V4.0A	6.1	Yes	Yes	Yes
Oracle RALLY (Development), V4.1	6.1	Yes	Yes	Yes
Oracle RALLY (MOTIF), V4.0A	6.1	Yes	No	No
Oracle RALLY (MOTIF), V4.1	6.1	Yes	Yes	Yes
Oracle Rdb, V5.1A	6.1	Yes		

(continued on next page)

Table D–1 (Cont.) Digital Layered Products Compatible with OpenVMS VAX Version 6.2

Product and Version	Operating System Support	DECnet Support		
	(Qualified)	Ph. IV	Syn	Full
Oracle Rdb, V6.0	6.1	Yes	Yes	No
Oracle Rdb, V6.0A	6.1	Yes	Yes	No
Oracle Rdb, V6.1	6.1	Yes	Yes	
Oracle Rdb, V6.1	6.1	Yes	Yes	No
Oracle Rdb (Interactive), V5.1A	6.1	Yes		
Oracle Rdb (Run-Time), V5.1A	6.1	Yes		
Oracle TRACE, V2.1A	6.1	Yes	Yes	
PATHWORKS for OpenVMS (Macintosh), V1.3	6.2	Yes	Yes	No
PATHWORKS for OpenVMS (NetWare), V1.0C	6.1	Yes	Yes	No
PATHWORKS for VMS, V5.0C	6.1	Yes	Yes	No
POLYCENTER Accounting Chargeback for OpenVMS, V2.1	6.1	Yes	Yes	Yes
POLYCENTER Capacity Planner, V3.0	6.1	Yes		
POLYCENTER Console Manager, V1.5A	6.1	N/A	N/A	N/A
POLYCENTER Console Manager, V1.6	6.2	N/A	N/A	N/A
POLYCENTER Extended LAN Manager for OpenVMS VAX, V1.4	6.1	Yes	Yes	No
POLYCENTER Fault Diagnostic Package, V1.4	6.1	Yes	Yes	No
POLYCENTER File Optimizer, V2.0	6.1	Yes	N/A	N/A
POLYCENTER File Optimizer, V2.1	6.2	Yes	N/A	N/A
POLYCENTER Framework, V1.4	6.1	Yes	Yes	No
POLYCENTER Framework Developer's Toolkit, V1.4	6.1	Yes	Yes	No
POLYCENTER Framework Historian Option, V1.4	6.1	Yes	Yes	No
POLYCENTER Framework Notification Option, V1.4	6.1	Yes	Yes	No
POLYCENTER HSM, V1.0A	6.1	Yes	N/A	N/A
POLYCENTER HSM, V1.1	6.1	Yes	N/A	N/A
POLYCENTER Media Library Manager for OpenVMS VAX, V1.1	6.1	Yes	Yes	Yes
POLYCENTER Network Manager 200, V1.4	6.1	Yes	Yes	No
POLYCENTER Network Manager 400, V2.4	6.1	Yes	Yes	No
POLYCENTER Network Statistics Option, V1.4	6.1	Yes	Yes	No
POLYCENTER Network Topology Option, V1.4	6.1	Yes	Yes	No
POLYCENTER Performance Advisor for OpenVMS, V2.1	6.1	Yes	Yes	Yes
POLYCENTER Performance Data Collector for OpenVMS, V2.1	6.1	Yes	Yes	Yes
POLYCENTER Saveset Manager for OpenVMS, V1.0A	6.1	Yes	No	No
POLYCENTER Saveset Manager for OpenVMS, V1.1	6.1	Yes	No	No
POLYCENTER Scheduler OpenVMS, V2.1B	6.1	Yes	Yes	No
POLYCENTER Scheduler OpenVMS (Agent), V2.1B	6.1	Yes	Yes	No
POLYCENTER Scheduler OpenVMS (Client), V2.1B	6.1	Yes	Yes	No

(continued on next page)

Table D–1 (Cont.) Digital Layered Products Compatible with OpenVMS VAX Version 6.2

Product and Version	Operating System Support	DECnet Support		
	(Qualified)	Ph. IV	Syn	Full
POLYCENTER Scheduler OpenVMS (Server), V2.1B	6.1	Yes	Yes	No
POLYCENTER Scheduler/Japanese for OpenVMS VAX, V2.1	6.1	Yes	No	No
POLYCENTER Security Compliance Manager for OpenVMS, V2.3A	6.1	Yes	Yes	No
POLYCENTER Security Intrusion Detector for VMS, V1.2	6.2	Yes	Yes	Yes
POLYCENTER Security Reporting Facility for OpenVMS, V2.1	6.1	Yes	Yes	No
POLYCENTER Sequential Media File System for OpenVMS VAX, V1.0	6.1	Yes	N/A	N/A
POLYCENTER Sequential Media File System for OpenVMS VAX, V1.1	6.1	Yes	N/A	N/A
POLYCENTER Software Distribution, V3.0	6.1	Yes	Yes	No
POLYCENTER Striping for OpenVMS VAX, V2.1	6.1	N/A	N/A	N/A
POLYCENTER System Census for OpenVMS VAX and ULTRIX (Agent), V1.3	6.1	Yes	Yes	No
POLYCENTER System Census for OpenVMS VAX and ULTRIX (Consolidator), V1.3	6.1	Yes	Yes	No
POLYCENTER System Census for OpenVMS VAX and ULTRIX (GUI), V1.3	6.1	Yes	Yes	No
POLYCENTER System Watchdog Agent for OpenVMS, V2.2	6.1	Yes	Yes	No
POLYCENTER System Watchdog Consolidator for OpenVMS, V2.2	6.1	Yes	Yes	No
POSIX for OpenVMS, V2.0	6.2	Yes	Yes	Yes
PrintServer Software, V5.1	6.1			
PrintServer Software (Base), V5.1	6.1			
PrintServer Software (Client), V5.1	6.1			
PrintServer Software (DECprint Supervisor), V5.1	6.1			
Reliable Transaction Router for OpenVMS VAX (Client), V2.2B	6.2	Yes	Yes	No
Reliable Transaction Router for OpenVMS VAX (Client), V2.2C	6.2	Yes	Yes	No
Reliable Transaction Router for OpenVMS VAX (Server), V2.2C	6.2	Yes	Yes	No
RMS Journaling for OpenVMS VAX, V6.2	6.2	Yes	Yes	Yes
Storage Library System for OpenVMS, V2.5	6.1	Yes	No	No
Storage Library System for OpenVMS (ACS), V2.5	6.1	Yes	No	No
Storage Library System for OpenVMS (Client), V2.5	6.1	Yes	No	No
StorageWorks Desktop Backup/Archiver Backup Server, V2.0	6.1	Yes	No	No
StorageWorks Desktop Backup/Archiver Backup Server, V2.0B	6.2	Yes	No	No
Storageworks RAID Software for OpenVMS, V1.0	6.2	N/A	N/A	N/A
Storageworks RAID Software for OpenVMS, V2.1	6.1	N/A	N/A	N/A
Storageworks RAID Software for OpenVMS, V2.2	6.2	N/A	N/A	N/A
TeamRoute for ALL–IN–1 OpenVMS VAX, V1.2	6.1	Yes	Yes	No
TeamRoute for ALL–IN–1 OpenVMS VAX, V1.3	6.1	Yes	Yes	No
TeamRoute for ALL–IN–1/Japanese for OpenVMS VAX, V1.1A	6.1	Yes		
TeamRoute for OpenVMS, V1.0	6.1	Yes		

(continued on next page)

Table D–1 (Cont.) Digital Layered Products Compatible with OpenVMS VAX Version 6.2

Product and Version	Operating System Support	DECnet Support		
	(Qualified)	Ph. IV	Syn	Full
TeamRoute/Dansk for ALL–IN–1, V1.1	6.1	Yes		
TeamRoute/Deutsch for ALL–IN–1, V1.1	6.1	Yes		
TeamRoute/Español for ALL–IN–1, V1.0	6.1	Yes		
TeamRoute/Français for ALL–IN–1, V1.1	6.1	Yes		
TeamRoute/Italiano for ALL–IN–1, V1.1	6.1	Yes		
TeamRoute/Japanese for OpenVMS, V1.0	6.1	Yes		
TeamRoute/Nederlands for ALL–IN–1, V1.1	6.1	Yes		
TeamRoute/Norsk for ALL–IN–1, V1.1	6.1	Yes		
TeamRoute/Suomi for ALL–IN–1, V1.1	6.1	Yes		
TeamRoute/Svenska for ALL–IN–1, V1.1	6.1	Yes		
VAX 2780/3780 Protocol Emulator, V1.8	6.1	Yes	Yes	No
VAX BASIC, V3.7	6.1	Yes	Yes	Yes
VAX COBOL for OpenVMS VAX, V5.2	6.1	Yes	Yes	Yes
VAX COBOL/Japanese for OpenVMS VAX, V5.2	6.1	Yes	Yes	Yes
VAX Distributed Name Service, V1.1A	6.1	Yes	Yes	Yes
VAX FMS, V2.4	6.1	N/A	N/A	N/A
VAX FMS (Run-Time), V2.4	6.1	N/A	N/A	N/A
VAX FMS/Japanese, V2.4	6.1	Yes	Yes	Yes
VAX FMS/Japanese (Run-Time), V2.4	6.1	Yes	Yes	Yes
VAX Message Router (Base System), V3.3A	6.1	Yes	Yes	No
VAX Message Router (Programmer's Kit), V3.3	6.1	Yes	Yes	No
VAX Message Router (VMSmail Gateway), V3.3A	6.1	Yes	Yes	No
VAX Message Router X.400 Gateway, V2.3	6.1	No	Yes	No
VAX Message Router/P Gateway, V1.3	6.1	Yes	Yes	No
VAX Message Router/S Gateway, V1.3	6.1	Yes	Yes	No
VAX TDMS, V1.9A	6.1	N/A	N/A	N/A
VAX TDMS (Run-Time), V1.9A	6.1	N/A	N/A	N/A
VAX–11/RSX, V2.5	6.1			
VAXcluster Software for OpenVMS VAX, V6.2	6.2	Yes	Yes	Yes
Volume Shadowing for OpenVMS VAX, V6.2	6.2	Yes	Yes	Yes

Table D–2 Digital Layered Products Compatible with OpenVMS Alpha Version 6.2

Product and Version	Operating System Support	DECnet Support		
	(Qualified)	Ph. IV	Syn	Full
ACMS Desktop, V2.0	6.1	Yes	Yes	No
ACMS Desktop/Japanese, V2.0	6.1			
ACMS for OpenVMS AXP (Development), V4.0	6.1	Yes	N/A	N/A
ACMS for OpenVMS AXP (Remote), V4.0	6.1	Yes	N/A	N/A
ACMS for OpenVMS AXP (Run-Time), V4.0	6.1	Yes	N/A	N/A
ALL–IN–1 Office Server Options for OpenVMS AXP, V3.1	6.1	Yes	Yes	No
ALL–IN–1 Teamlinks Connection Package for OpenVMS AXP, V2.1	6.1	Yes	Yes	No
ALL–IN–1 Teamlinks Connection Package for OpenVMS AXP, V2.5	6.1	Yes	Yes	No
ALL–IN–1/Dansk Office Server Options for OpenVMS AXP and VAX, V3.1	6.1	Yes	Yes	No
ALL–IN–1/Deutsch Office Server Options for Open VMS AXP and VAX, V3.1	6.1	Yes	Yes	No
ALL–IN–1/Español Office Server Options for OpenVMS AXP and VAX, V3.1	6.1	Yes	Yes	No
ALL–IN–1/Français Office Server Options for OpenVMS AXP and VAX, V3.1	6.1	Yes	Yes	No
ALL–IN–1/Italiano Office Server Options for OpenVMS AXP and VAX, V3.1	6.1	Yes	Yes	No
ALL–IN–1/Japanese Office Server Options for OpenVMS AXP and VAX, V3.1	6.1			
ALL–IN–1/Nederlands Office Server Options for OpenVMS AXP and VAX, V3.1	6.1	Yes	Yes	No
ALL–IN–1/Norsk Office Server Options for OpenVMS AXP and VAX, V3.1	6.1	Yes	Yes	No
ALL–IN–1/Portugûes Office Server Options for OpenVMS AXP and VAX, V3.1	6.1	Yes	Yes	No
ALL–IN–1/Suomi Office Server Options for OpenVMS AXP and VAX, V3.1	6.1	Yes	Yes	No
ALL–IN–1/Svenska Office Server Options for OpenVMS AXP and VAX, V3.1	6.1	Yes	Yes	No
AssetWORKS for OpenVMS AXP (Agent), V1.0	6.2			
AssetWORKS for OpenVMS AXP (Server), V1.0	6.2			
BASEstar Classic for OpenVMS AXP (Development), V3.3	6.1	Yes	Yes	No
BASEstar Classic for OpenVMS AXP (Run-Time), V3.3	6.1	Yes	Yes	No
BASEstar Graphics Enabler for OpenVMS AXP (Development), V2.0	6.1	Yes	No	
BASEstar Graphics Enabler for OpenVMS AXP (Run-Time), V2.0	6.1	Yes	No	
BASEstar Open Server for OpenVMS AXP (Client), V2.0	6.1	N/A	N/A	N/A
BASEstar Open Server for OpenVMS AXP (Development), V2.0	6.1	N/A	N/A	N/A
BASEstar Open Server for OpenVMS AXP (Run-Time), V2.0	6.1	N/A	N/A	N/A
DEC Ada for OpenVMS Alpha Systems, V3.0A	6.1	Yes	Yes	No
DEC Ada for OpenVMS Alpha Systems, V3.2	6.2	Yes	Yes	No
DEC Ada for Professional Development Option for OpenVMS AXP Systems, V3.2	6.2	Yes	Yes	No
DEC Availability Manager for Distributed Systems, V6.1	6.1			
DEC BASIC for OpenVMS AXP, V1.0	6.1			
DEC BASIC for OpenVMS AXP, V1.1	6.2	N/A	Yes	Yes
DEC C for OpenVMS AXP Systems, V1.3A	6.1	N/A	N/A	N/A
DEC C for OpenVMS AXP Systems, V4.0	6.1	N/A	N/A	N/A

(continued on next page)

Table D–2 (Cont.) Digital Layered Products Compatible with OpenVMS Alpha Version 6.2

Product and Version	Operating System Support	DECnet Support		
	(Qualified)	Ph. IV	Syn	Full
DEC C for OpenVMS AXP Systems, V4.1	6.1	N/A	N/A	N/A
DEC C for OpenVMS AXP Systems, V5.0	6.2	N/A	N/A	N/A
DEC C++ for OpenVMS AXP Systems, V1.3A	6.1	N/A	N/A	N/A
DEC C++ for OpenVMS AXP Systems, V5.0	6.2	N/A	N/A	N/A
DEC COBOL for OpenVMS Alpha Systems, V2.0	6.1	Yes	No	No
DEC COBOL for OpenVMS Alpha Systems, V2.1	6.2	Yes	No	No
DEC COBOL for OpenVMS Alpha Systems, V2.2	6.2	Yes	No	No
DEC COBOL/Japanese for OpenVMS AXP Systems, V2.0	6.1			
DEC COBOL/Japanese for OpenVMS AXP Systems, V2.1	6.1			
DEC Code Management System (CMS) for OpenVMS AXP Systems, V3.6	6.1	Yes	No	No
DEC Code Management System (CMS) for OpenVMS AXP Systems, V3.7	6.2	Yes	No	No
DEC Data Distributor Version for OpenVMS AXP, V6.0	6.1	Yes	Yes	No
DEC DATATRIEVE for OpenVMS AXP Systems, V6.1B	6.1	Yes	Yes	No
DEC DATATRIEVE for OpenVMS AXP Systems, V7.0	6.1	Yes	Yes	Yes
DEC DATATRIEVE/Japanese for OpenVMS AXP Systems, V6.1	6.1	Yes	Yes	No
DEC DB Integrator for OpenVMS AXP, V1.0	6.1	Yes	No	No
DEC DB Integrator for OpenVMS AXP, V1.0B	6.1	Yes		
DEC DB Integrator for OpenVMS AXP, V3.1	6.1	Yes		
DEC DB Integrator Gateway for Custom Drivers for OpenVMS AXP, V3.0	6.1			
DEC DB Integrator Gateway for Custom Drivers for OpenVMS AXP, V3.0B	6.1			
DEC DB Integrator Gateway for Custom Drivers for OpenVMS AXP, V3.0C	6.1			
DEC DB Integrator Gateway for DB2 for OpenVMS AXP, V3.0	6.1			
DEC DB Integrator Gateway for DB2 for OpenVMS AXP, V3.0B	6.1			
DEC DB Integrator Gateway for DB2 for OpenVMS AXP, V3.1	6.1			
DEC DB Integrator Gateway for DBMS for OpenVMS AXP, V1.0	6.1	Yes	Yes	No
DEC DB Integrator Gateway for DSM, V1.0B	6.1	Yes		
DEC DB Integrator Gateway for EDA/SQL, V3.1	6.1	Yes		
DEC DB Integrator Gateway for ORACLE for OpenVMS AXP, V3.0B	6.1			
DEC DB Integrator Gateway for ORACLE for OpenVMS AXP, V3.1	6.1			
DEC DB Integrator Gateway for RMS for OpenVMS AXP, V3.0	6.1			
DEC DB Integrator Gateway for RMS for OpenVMS AXP, V3.0B	6.1			
DEC DB Integrator Gateway for RMS for OpenVMS AXP, V3.0C	6.1			
DEC DB Integrator Gateway for Sequelink for OpenVMS AXP, V3.1	6.1			
DEC DB Integrator Gateway for SYBASE for OpenVMS AXP, V1.0B	6.1			
DEC DB Integrator Gateway for SYBASE for OpenVMS AXP, V3.1	6.1			
DEC Distributed Computing Services for OpenVMS, V1.0	6.1			

(continued on next page)

Table D–2 (Cont.) Digital Layered Products Compatible with OpenVMS Alpha Version 6.2

Product and Version	Operating System Support	DECnet Support		
	(Qualified)	Ph. IV	Syn	Full
DEC Distributed Queueing Service for OpenVMS AXP, V1.3	6.2	Yes	Yes	Yes
DEC FMS for OpenVMS AXP (Development), V2.4	6.2	N/A	N/A	N/A
DEC FMS for OpenVMS AXP (Run-Time), V2.4	6.2	N/A	N/A	N/A
DEC FMS/Hangul for OpenVMS AXP (Development), V2.4	6.1	N/A	N/A	N/A
DEC FMS/Hangul for OpenVMS AXP (Run-Time), V2.4	6.1	N/A	N/A	N/A
DEC FMS/Hebrew for OpenVMS AXP (Development), V2.4	6.1	N/A	N/A	N/A
DEC FMS/Hebrew for OpenVMS AXP (Run-Time), V2.4	6.1	N/A	N/A	N/A
DEC FMS/Japanese for OpenVMS AXP (Development), V2.4	6.1	N/A	N/A	N/A
DEC FMS/Japanese for OpenVMS AXP (Run-Time), V2.4	6.1	N/A	N/A	N/A
DEC Fortran for OpenVMS Systems, V6.2	6.1	Yes	Yes	Yes
DEC Fortran for OpenVMS Systems, V6.3	6.2	Yes	Yes	Yes
DEC Fortran-90 for OpenVMS Systems, V2.0	6.2	Yes	Yes	Yes
DEC GKS for OpenVMS AXP Systems (Development), V5.3	6.1	Yes	Yes	Yes
DEC GKS for OpenVMS AXP Systems (Development), V6.0	6.1	Yes	Yes	Yes
DEC GKS for OpenVMS AXP Systems (Run-Time), V5.3	6.1	Yes	Yes	Yes
DEC GKS for OpenVMS AXP Systems (Run-Time), V6.0	6.1	Yes	Yes	Yes
DEC GKS/Japanese for OpenVMS AXP (Development), V5.2A	6.1			
DEC GKS/Japanese for OpenVMS AXP (Development), V5.3	6.1	Yes	Yes	Yes
DEC GKS/Japanese for OpenVMS AXP (Run-Time), V5.2A	6.1			
DEC GKS/Japanese for OpenVMS AXP (Run-Time), V5.3	6.1	Yes	Yes	Yes
DEC Language Sensitive Editor/Source Code Analyzer for OpenVMS Systems, V4.2	6.1	Yes	No	No
DEC Language Sensitive Editor/Source Code Analyzer for OpenVMS Systems, V4.3	6.2	Yes	No	No
DEC MAILWorks for OpenVMS, V1.3A	6.1	Yes	Yes	No
DEC Module Management System (MMS) for OpenVMS AXP Systems, V2.7	6.1	Yes	No	No
DEC Module Management System (MMS) for OpenVMS AXP Systems, V3.0	6.2	Yes	No	No
DEC Notes for OpenVMS AXP Systems, V2.5	6.1			
DEC Open3D for OpenVMS AXP, V2.3	6.1	Yes	Yes	Yes
DEC Open3D for OpenVMS AXP, V2.4	6.1	Yes	Yes	Yes
DEC Open3D for OpenVMS AXP, V2.5	6.1	Yes	Yes	Yes
DEC Open3D for OpenVMS AXP, V2.6	6.2	Yes	Yes	Yes
DEC OPS5 for OpenVMS, V4.0A	6.1	Yes	Yes	Yes
DEC OPS5/Japanese for OpenVMS AXP, V4.0A	6.1	Yes	Yes	Yes
DEC Optical Storage Desktop Software (OSDS) for OpenVMS AXP, V3.2	6.1	N/A	N/A	N/A
DEC Optical Storage Management Software (OSMS) for OpenVMS AXP, V3.2	6.1	N/A	N/A	N/A
DEC OSDS-Optical Storage Desktop Software for OpenVMS AXP, V3.2	6.1	N/A	N/A	N/A

(continued on next page)

Table D–2 (Cont.) Digital Layered Products Compatible with OpenVMS Alpha Version 6.2

Product and Version	Operating System Support	DECnet Support		
	(Qualified)	Ph. IV	Syn	Full
DEC Pascal for OpenVMS AXP Systems, V5.2	6.1	Yes	Yes	Yes
DEC Pascal for OpenVMS AXP Systems, V5.3	6.1	Yes	Yes	Yes
DEC Performance and Coverage Analyzer (PCA) for OpenVMS AXP, V4.2	6.1	Yes	No	No
DEC Performance and Coverage Analyzer (PCA) for OpenVMS AXP, V4.3	6.2	Yes	No	No
DEC PHIGS for OpenVMS AXP Systems (Development), V2.5A	6.1			
DEC PHIGS for OpenVMS AXP Systems (Development), V2.6	6.1	Yes	Yes	Yes
DEC PHIGS for OpenVMS AXP Systems (Development), V3.0	6.1	Yes	Yes	Yes
DEC PHIGS for OpenVMS AXP Systems (Development), V3.0B	6.1	Yes	Yes	Yes
DEC PHIGS for OpenVMS AXP Systems (Development), V3.1	6.2	Yes	Yes	Yes
DEC PHIGS for OpenVMS AXP Systems (Run-Time), V2.5A	6.1			
DEC PHIGS for OpenVMS AXP Systems (Run-Time), V2.6	6.1	Yes	Yes	Yes
DEC PHIGS for OpenVMS AXP Systems (Run-Time), V3.0	6.1	Yes	Yes	Yes
DEC PHIGS for OpenVMS AXP Systems (Run-Time), V3.0B	6.1	Yes	Yes	Yes
DEC PHIGS for OpenVMS AXP Systems (Run-Time), V3.1	6.2	Yes	Yes	Yes
DEC PHIGS/Japanese for OpenVMS AXP (Development), V2.5	6.1	Yes	Yes	Yes
DEC PHIGS/Japanese for OpenVMS AXP (Run-Time), V2.5	6.1	Yes	Yes	Yes
DEC Rdb/Hangul for OpenVMS/Hangul AXP, V6.0	6.1	Yes	Yes	No
DEC Rdb/Hanyu for OpenVMS/Hanyu AXP, V6.0	6.1	Yes	Yes	No
DEC Rdb/Japanese for OpenVMS AXP, V6.0	6.1			
DEC SNA 3270 Application Services (Development), V1.1	6.1	Yes	Yes	No
DEC SNA 3270 Application Services (Run-Time), V1.1	6.1	Yes	Yes	No
DEC SoftWindows/SoftPC for OpenVMS, V5.0	6.1		Yes	Yes
DEC TCP/IP Services for OpenVMS AXP, V3.1	6.1			
DEC TCP/IP Services for OpenVMS AXP, V3.2	6.1			
DEC TCP/IP Services for OpenVMS AXP, V3.3	6.1			
DEC TCP/IP Services/Japanese for OpenVMS AXP, V3.0	6.1			
DEC TCP/IP Services/Japanese for OpenVMS AXP, V3.1	6.1			
DEC TCP/IP Services/Japanese for OpenVMS AXP, V3.2	6.1			
DEC Test Manager for OpenVMS AXP Systems, V3.5	6.1			
DEC Test Manager for OpenVMS AXP Systems, V3.6	6.2	Yes	No	No
DEC X.25 Client for OpenVMS AXP Systems, V1.1	6.2	Yes	N/A	N/A
DEC X.500 Directory Services (Administration Facility), V1.3	6.1	N/A	Yes	Yes
DEC X.500 Directory Services (Administration Facility), V2.0	6.1	N/A	Yes	Yes
DEC X.500 Directory Services (Administration Facility), V2.0A	6.2	N/A	Yes	Yes
DEC X.500 Directory Services (Server), V1.3	6.1	N/A	Yes	Yes
DEC X.500 Directory Services (Server), V2.0	6.1	N/A	Yes	Yes

(continued on next page)

Table D–2 (Cont.) Digital Layered Products Compatible with OpenVMS Alpha Version 6.2

Product and Version	Operating System Support	DECnet Support		
	(Qualified)	Ph. IV	Syn	Full
DEC X.500 Directory Services (Server), V2.0A	6.2	N/A	Yes	Yes
DEC X.500 Directory Services (Starter Kit), V1.3	6.1	N/A	Yes	Yes
DEC X.500 Directory Services (Starter Kit), V2.0	6.1	N/A	Yes	Yes
DEC X.500 Directory Services (Starter Kit), V2.0A	6.2	N/A	Yes	Yes
DEC/EDI for OpenVMS AXP, V2.0	6.1	Yes	Yes	No
DEC/EDI for OpenVMS AXP, V2.1	6.2	Yes	Yes	No
DECADMIRE for OpenVMS AXP, V2.0A	6.1	N/A	N/A	N/A
DECADMIRE for OpenVMS AXP, V2.1A	6.2	N/A	N/A	N/A
DECADMIRE/Japanese for OpenVMS, V2.0	6.1	N/A	N/A	N/A
DECdfs for OpenVMS Systems, V2.0	6.1	Yes	Yes	No
DECdfs for OpenVMS Systems, V2.1	6.2	Yes	Yes	Yes
DECdocument for OpenVMS AXP Systems, V2.3	6.2	N/A	N/A	N/A
DECforms /Hangul for OpenVMS AXP (Development), V1.4C	6.1	N/A	N/A	N/A
DECforms /Hangul for OpenVMS AXP (Run-Time), V1.4A	6.1	N/A	N/A	N/A
DECforms /Hangul for OpenVMS AXP (Run-Time), V1.4C	6.1	N/A	N/A	N/A
DECforms /Hanyu for OpenVMS AXP (Development), V1.4C	6.1	N/A	N/A	N/A
DECforms /Hanyu for OpenVMS AXP (Run-Time), V1.4A	6.1	N/A	N/A	N/A
DECforms /Hanyu for OpenVMS AXP (Run-Time), V1.4C	6.1	N/A	N/A	N/A
DECforms /Japanese for OpenVMS AXP (Run-Time), V1.4	6.1	N/A	N/A	N/A
DECforms /Thai for OpenVMS AXP (Run-Time), V1.4A	6.1	N/A	N/A	N/A
DECforms for OpenVMS AXP (Development), V1.4C	6.2			
DECforms for OpenVMS AXP (Development), V1.4D	6.2	N/A	N/A	N/A
DECforms for OpenVMS AXP (Development), V2.1	6.2	N/A	N/A	N/A
DECforms for OpenVMS AXP (Run-Time), V1.4C	6.2			
DECforms for OpenVMS AXP (Run-Time), V1.4D	6.2	N/A	N/A	N/A
DECforms for OpenVMS AXP (Run-Time), V2.1	6.2	N/A	N/A	N/A
DECmessageQ for OpenVMS AXP (Run-Time), V2.0B	6.1			
DECmessageQ for OpenVMS AXP (Run-Time), V2.0C	6.1	Yes	No	No
DECmessageQ for OpenVMS AXP (Run-Time), V3.0	6.2	Yes	Yes	No
DECmessageQ for OpenVMS AXP Systems (Development), V2.0B	6.1			
DECmessageQ for OpenVMS AXP Systems (Development), V2.0C	6.1	Yes	No	No
DECmessageQ for OpenVMS AXP Systems (Development), V3.0	6.2	Yes	Yes	No
DECmigrate for OpenVMS AXP Systems, V1.1	6.1	N/A	N/A	N/A
DECmigrate for OpenVMS AXP Systems, V1.1A	6.2	N/A	N/A	N/A
DECnet for OpenVMS AXP, V5.7	6.1			
DECnet for OpenVMS AXP, V5.8	6.1	Yes	Yes	Yes

(continued on next page)

Table D–2 (Cont.) Digital Layered Products Compatible with OpenVMS Alpha Version 6.2

Product and Version	Operating System Support	DECnet Support		
	(Qualified)	Ph. IV	Syn	Full
DECnet for OpenVMS AXP, V6.1	6.1			
DECnet for OpenVMS AXP, V6.1	6.1	N/A	Yes	Yes
DECnet for OpenVMS AXP, V6.2	6.2	N/A	Yes	Yes
DECnet SNA 3270 Data Stream Programming Interface for OpenVMS, V1.4	6.1	Yes	Yes	No
DECnet SNA 3270 Data Stream Programming Interface for OpenVMS/Japanese, V1.4	6.1			
DECnet SNA 3270 Terminal Emulator for OpenVMS, V1.5	6.1	Yes	Yes	No
DECnet SNA 3270 Terminal Emulator/Japanese for OpenVMS, V1.5A	6.1	Yes	No	No
DECnet SNA APPC/LU6.2 Programming Interface for OpenVMS, V2.2	6.1	Yes	Yes	No
DECnet SNA Application Programming Interface for OpenVMS, V2.3	6.1	Yes	Yes	No
DECnet SNA Data Transfer Facility for OpenVMS (Server), V3.1	6.1	Yes	Yes	No
DECnet SNA Data Transfer Facility for OpenVMS (Utility), V3.1	6.1	Yes	Yes	No
DECnet SNA DTF/Japanese for OpenVMS (Utilities), V3.1	6.1			
DECnet SNA DTF/Japanese for OpenVMS AXP (Server), V3.1	6.1			
DECnet SNA Gateway for Channel Transport, V2.1	6.1	Yes	Yes	No
DECnet SNA Gateway for Synchronous Transport, V1.2	6.1	Yes	Yes	No
DECnet SNA Printer Emulator for OpenVMS, V1.2	6.1	Yes	Yes	No
DECnet SNA Printer Emulator/Japanese for OpenVMS AXP, V1.2A	6.1			
DECnet SNA Remote Job Entry for OpenVMS, V1.4	6.1	Yes	Yes	No
DECnet SNA Remote Job Entry/Japanese for OpenVMS, V1.4A	6.1	Yes	Yes	No
DECnet/OSI for OpenVMS Alpha AXP, V5.7	6.1			
DECnet/OSI for OpenVMS Alpha AXP, V5.8	6.1	Yes	Yes	Yes
DECnet/OSI for OpenVMS Alpha AXP, V6.1	6.1			
DECnet/OSI for OpenVMS Alpha AXP, V6.1	6.1	N/A	Yes	Yes
DECnet/OSI for OpenVMS Alpha AXP, V6.2	6.2	N/A	Yes	Yes
DEComni API, V2.2	6.1	N/A	N/A	N/A
DEComni MMS, V2.2	6.1			
DECosap/AP for OpenVMS AXP, V2.0	6.1	N/A	N/A	N/A
DECosap/H1 for OpenVMS AXP, V2.0	6.1	N/A	N/A	N/A
DECprint Supervisor for OpenVMS AXP, V1.1	6.1	Yes	No	No
DECprint Supervisor for OpenVMS AXP, V1.1A	6.1	Yes	No	No
DECprint Supervisor for OpenVMS AXP, V1.2	6.2	Yes	No	No
DECprint Supervisor/Japanese for OpenVMS, V1.0	6.1			
DECprint Supervisor/Japanese for OpenVMS, V1.1	6.1			
DECram for OpenVMS, V2.1	6.2	N/A	N/A	N/A
DECserver Network Access Software, V1.3	6.1			

(continued on next page)

Table D–2 (Cont.) Digital Layered Products Compatible with OpenVMS Alpha Version 6.2

Product and Version	Operating System Support	DECnet Support		
	(Qualified)	Ph. IV	Syn	Full
DECset for OpenVMS AXP Systems, V11.2	6.1	Yes	No	No
DECset for OpenVMS AXP Systems, V12.0	6.2			
DECwindows DECnet SNA 3270 Terminal Emulator for OpenVMS, V2.0	6.1	Yes	Yes	No
DECwindows DECnet SNA 3270 Terminal Emulator for OpenVMS/Japanese, V2.0A	6.1	Yes	No	No
DECwindows Motif for OpenVMS AXP, V1.2	6.2	Yes	Yes	Yes
DECwindows Motif for OpenVMS AXP, V1.2-3	6.2	Yes	Yes	Yes
DECwindows Motif OpenVMS User Interface/Cesky, V1.1A	6.1	Yes	Yes	Yes
DECwindows Motif OpenVMS User Interface/Cesky, V1.2	6.1	Yes	Yes	Yes
DECwindows Motif OpenVMS User Interface/Deutsch, V1.1	6.1			
DECwindows Motif OpenVMS User Interface/Deutsch, V1.2	6.1	Yes	Yes	Yes
DECwindows Motif OpenVMS User Interface/Español, V1.1	6.1			
DECwindows Motif OpenVMS User Interface/Español, V1.2	6.1	Yes	Yes	Yes
DECwindows Motif OpenVMS User Interface/Français, V1.1	6.1			
DECwindows Motif OpenVMS User Interface/Français, V1.2	6.1	Yes	Yes	Yes
DECwindows Motif OpenVMS User Interface/Italiano, V1.1	6.1			
DECwindows Motif OpenVMS User Interface/Italiano, V1.2	6.1	Yes	Yes	Yes
DECwindows Motif OpenVMS User Interface/Magyar, V1.1A	6.1			
DECwindows Motif OpenVMS User Interface/Magyar, V1.2	6.1	Yes	Yes	Yes
DECwindows Motif OpenVMS User Interface/Polski, V1.1	6.1			
DECwindows Motif OpenVMS User Interface/Polski, V1.2	6.1	Yes	Yes	Yes
DECwindows Motif OpenVMS User Interface/Russkij, V1.1A	6.1			
DECwindows Motif OpenVMS User Interface/Russkij, V1.2	6.1	Yes	Yes	Yes
DECwindows Motif OpenVMS User Interface/Slovensky, V1.1A	6.1			
DECwindows Motif OpenVMS User Interface/Slovensky, V1.2	6.1	Yes	Yes	Yes
DECwindows Motif OpenVMS User Interface/Svenska, V1.1	6.1			
DECwindows Motif OpenVMS User Interface/Svenska, V1.2	6.1	Yes	Yes	Yes
DECwindows Motif/Hangul for OpenVMS AXP, V1.2	6.1	Yes	Yes	Yes
DECwindows Motif/Hanyu for OpenVMS AXP, V1.2	6.1	Yes	Yes	Yes
DECwindows Motif/Hanzi for OpenVMS AXP, V1.2	6.1	Yes	Yes	Yes
DECwindows Motif/Hebrew for OpenVMS AXP, V1.1	6.1			
DECwindows Motif/Hebrew for OpenVMS AXP, V1.2	6.1	Yes	Yes	No
DECwindows Motif/Japanese for OpenVMS AXP, V1.2	6.1			
DECwindows Motif/Thai for OpenVMS AXP, V1.2	6.1	Yes	Yes	Yes
DECwrite for OpenVMS AXP, V3.0	6.1	N/A	N/A	N/A
DECwrite/Deutsch for OpenVMS AXP, V3.0	6.1	N/A	N/A	N/A

(continued on next page)

Table D–2 (Cont.) Digital Layered Products Compatible with OpenVMS Alpha Version 6.2

Product and Version	Operating System Support	DECnet Support		
	(Qualified)	Ph. IV	Syn	Full
DECwrite/Français for OpenVMS AXP, V3.0	6.1	N/A	N/A	N/A
Digital Distributed Computing Environment (DCE), V1.3	6.2			
Digital Extended Math Library for OpenVMS AXP (Development), V2.7	6.1	N/A		
Digital Extended Math Library for OpenVMS AXP (Development), V2.8	6.2	N/A		
Digital Extended Math Library for OpenVMS AXP (Run-Time), V2.7	6.1	N/A		
Digital Extended Math Library for OpenVMS AXP (Run-Time), V2.8	6.2	N/A		
Digital NAS Base Server 200 for OpenVMS AXP Systems, V5.0	6.1	N/A	N/A	N/A
Digital NAS Base Server 200 for OpenVMS AXP Systems, V5.0A	6.1	N/A	N/A	N/A
Digital NAS Base Server 200 for OpenVMS AXP Systems, V6.0	6.1	N/A	N/A	N/A
Digital NAS Client 150 for OpenVMS AXP Systems, V5.1	6.1	N/A	N/A	N/A
Digital NAS Client 150 for OpenVMS AXP Systems, V5.2	6.1	N/A	N/A	N/A
Digital NAS Client 150 for OpenVMS AXP Systems, V6.0	6.1	N/A	N/A	N/A
Digital NAS Client 250 for OpenVMS AXP Workstations, V5.0	6.1	N/A	N/A	N/A
Digital NAS Client 250 for OpenVMS AXP Workstations, V6.0	6.1	N/A	N/A	N/A
Digital NAS Production Server 400 for OpenVMS AXP Systems, V5.0	6.1	N/A	N/A	N/A
Digital NAS Production Server 400 for OpenVMS AXP Systems, V6.0	6.1	N/A	N/A	N/A
Digital NAS Server 300 for OpenVMS AXP Systems, V5.0	6.1	N/A	N/A	N/A
Digital NAS Server 300 for OpenVMS AXP Systems, V6.0	6.1	N/A	N/A	N/A
DSM for OpenVMS AXP Systems, V6.3A	6.1			
DSM for OpenVMS AXP Systems, V6.3B	6.1	Yes	No	No
DSM for OpenVMS AXP Systems, V6.3C	6.1	Yes	No	No
DSM for OpenVMS AXP Systems, V6.3D	6.2	Yes	No	No
DSM/Japanese for OpenVMS AXP Systems, V6.2C	6.1			
DSM/Japanese for OpenVMS AXP Systems, V6.3	6.1			
International Lexicons for OpenVMS AXP (American Business), V1.1	6.1			
International Lexicons for OpenVMS AXP (American English), V1.1	6.1			
International Lexicons for OpenVMS AXP (American Medical), V1.1	6.1			
International Lexicons for OpenVMS AXP (British English), V1.1	6.1			
International Lexicons for OpenVMS AXP (Danish), V1.1	6.1			
International Lexicons for OpenVMS AXP (Dutch), V1.1	6.1			
International Lexicons for OpenVMS AXP (French), V1.1	6.1			
International Lexicons for OpenVMS AXP (German), V1.1	6.1			
International Lexicons for OpenVMS AXP (Italian), V1.1	6.1			
International Lexicons for OpenVMS AXP (Norwegian), V1.1	6.1			
International Lexicons for OpenVMS AXP (Portuguese), V1.1	6.1			
International Lexicons for OpenVMS AXP (Spanish), V1.1	6.1			

(continued on next page)

Table D–2 (Cont.) Digital Layered Products Compatible with OpenVMS Alpha Version 6.2

Product and Version	Operating System Support	DECnet Support		
	(Qualified)	Ph. IV	Syn	Full
International Lexicons for OpenVMS AXP (Swedish), V1.1	6.1			
KAP for DEC C for OpenVMS AXP, V1.6	6.1	N/A	N/A	N/A
KAP for DEC C for OpenVMS AXP, V1.6A	6.2	N/A	N/A	N/A
KAP for DEC Fortran for OpenVMS, V2.1	6.2	N/A	N/A	N/A
MACRO-64 Assembler for OpenVMS AXP Systems, V1.1	6.1			
MAILbus 400 Application Program Interface for OpenVMS, V1.4	6.1			
MAILbus 400 Message Transfer Agent for OpenVMS, V1.4	6.1			
MUXserver 320/380/90 Remote Terminal Server, V2.0	6.2	Yes	No	No
ObjectBroker for OpenVMS AXP (Development), V2.5A	6.1	Yes	No	No
ObjectBroker for OpenVMS AXP (Run-Time), V2.5A	6.1	Yes	No	No
Oracle CDD/Administrator for OpenVMS Systems, V1.2	6.1	Yes	No	No
Oracle CDD/Repository for OpenVMS Systems, V5.3	6.1	Yes	No	No
Oracle CDD/Repository for OpenVMS Systems, V6.1	6.1	Yes	No	No
Oracle CODASYL DBMS (Development), V6.0A	6.2	Yes	Yes	No
Oracle CODASYL DBMS (Development), V6.1	6.2	Yes	Yes	No
Oracle CODASYL DBMS (Run-Time), V6.0A	6.2	Yes	Yes	No
Oracle CODASYL DBMS (Run-Time), V6.1	6.2	Yes	Yes	No
Oracle Expert for Rdb, V2.1A	6.1	N/A	N/A	N/A
Oracle Expert for Rdb, V3.0	6.1	N/A	N/A	N/A
Oracle Graphical Schema Editor for Rdb, OpenVMS, V2.1A	6.1	N/A	N/A	N/A
Oracle InstantSQL for Rdb, V2.1A	6.1	N/A	N/A	N/A
Oracle RALLY, V4.0	6.1			
Oracle RALLY, V4.0A	6.1			
Oracle RALLY (Development), V4.0	6.1			
Oracle RALLY (Development), V4.0A	6.1			
Oracle RALLY (MOTIF), V4.0	6.1	Yes	No	No
Oracle RALLY (MOTIF), V4.0A	6.1	Yes	No	No
Oracle RALLY (Run-Time), V4.0	6.1			
Oracle RALLY (Run-Time), V4.0A	6.1			
Oracle Rdb (Multiversion), V5.1A	6.1			
Oracle Rdb (Multiversion), V6.0	6.1	Yes	Yes	No
Oracle Rdb (Multiversion), V6.0A	6.2	Yes	Yes	No
Oracle Rdb (Multiversion), V6.1	6.2	Yes	Yes	No
Oracle Rdb (Standard), V5.1A	6.1			
Oracle Rdb (Standard), V6.0	6.1	Yes	Yes	No
Oracle Rdb (Standard), V6.0A	6.2	Yes	Yes	No

(continued on next page)

Table D–2 (Cont.) Digital Layered Products Compatible with OpenVMS Alpha Version 6.2

Product and Version	Operating System Support	DECnet Support		
	(Qualified)	Ph. IV	Syn	Full
Oracle Rdb (Standard), V6.1	6.2	Yes	Yes	No
Oracle TRACE for OpenVMS, V2.1A	6.1	Yes	Yes	
OSI Application Developer's Toolkit for OpenVMS AXP, V2.2	6.1			
PATHWORKS for OpenVMS, V5.0A	6.1	Yes	Yes	No
PATHWORKS for OpenVMS, V5.0B	6.1	Yes	Yes	No
PATHWORKS for OpenVMS, V5.0C	6.2	Yes	Yes	No
PATHWORKS for OpenVMS (Macintosh), V1.3	6.2	Yes	Yes	No
PATHWORKS for OpenVMS (Netware), V1.0C	6.2	Yes	Yes	No
PL/I for OpenVMS AXP Systems, V4.0	6.2			
POLYCENTER Accounting Chargeback for OpenVMS, V2.1	6.1	Yes	Yes	Yes
POLYCENTER Archive/Backup for OpenVMS, V2.4A	6.1	Yes	No	No
POLYCENTER Archive/Backup for OpenVMS, V2.5B	6.2	Yes	No	No
POLYCENTER Capacity Planner for OpenVMS, V2.1	6.1	Yes	N/A	N/A
POLYCENTER Capacity Planner for OpenVMS, V2.1A	6.1	Yes	N/A	N/A
POLYCENTER Capacity Planner for OpenVMS, V3.0	6.1	Yes	N/A	N/A
POLYCENTER Capacity Planner for OpenVMS, V3.0A	6.1	Yes	N/A	N/A
POLYCENTER Console Manager for OpenVMS, V1.1	6.1	N/A	N/A	N/A
POLYCENTER Console Manager for OpenVMS, V1.5	6.1	N/A	N/A	N/A
POLYCENTER File Optimizer for OpenVMS, V2.0	6.2	Yes	No	No
POLYCENTER File Optimizer for OpenVMS, V2.1	6.2	N/A	N/A	N/A
POLYCENTER Hierarchical Storage Management for OpenVMS, V1.0	6.1	Yes	N/A	N/A
POLYCENTER Hierarchical Storage Management for OpenVMS, V1.0A	6.1	Yes	N/A	N/A
POLYCENTER Hierarchical Storage Management for OpenVMS, V1.1	6.2	Yes	N/A	N/A
POLYCENTER Installation for OpenVMS, V1.0	6.1			
POLYCENTER Installation for OpenVMS, V1.1	6.2			
POLYCENTER Performance Advisor for OpenVMS, V2.1	6.1	Yes	Yes	Yes
POLYCENTER Performance Data Collector for OpenVMS, V2.1	6.1	Yes	Yes	Yes
POLYCENTER Save Set Manager for OpenVMS, V1.1	6.2			
POLYCENTER Scheduler for OpenVMS, V2.1B	6.1	Yes	Yes	Yes
POLYCENTER Scheduler for OpenVMS (Agent), V2.1B	6.1	Yes	Yes	Yes
POLYCENTER Security Compliance Manager for OpenVMS, V2.3A	6.1	Yes	Yes	No
POLYCENTER Security Intrusion Detector for OpenVMS AXP, V1.2	6.1			
POLYCENTER Sequential Media Filesystem for OpenVMS VAX/Alpha, V1.1	6.2	N/A	N/A	N/A
POLYCENTER Software Distribution, V3.0	6.1	Yes	Yes	No
POLYCENTER Software Distribution, V3.1	6.2	Yes	Yes	No
POLYCENTER System Census Agent for OpenVMS, V1.3	6.1	Yes	Yes	No

(continued on next page)

Table D–2 (Cont.) Digital Layered Products Compatible with OpenVMS Alpha Version 6.2

Product and Version	Operating System Support	DECnet Support		
	(Qualified)	Ph. IV	Syn	Full
POLYCENTER System Census Consolidator for OpenVMS, V1.3	6.1	Yes	Yes	No
POLYCENTER System Census GUI for OpenVMS, V1.3	6.1	Yes	Yes	No
POLYCENTER System Watchdog (Agent), V2.2	6.2	Yes	Yes	No
POLYCENTER System Watchdog (Consolidator), V2.2	6.2	Yes	Yes	No
POSIX for OpenVMS AXP, V2.0	6.2	Yes	Yes	Yes
PrintServer Software, V5.0	6.2	Yes	No	No
PrintServer Software, V5.0B	6.2	Yes	No	No
PrintServer/Japanese Software for OpenVMS, V5.0	6.2	Yes	No	No
Reliable Transaction Router for OpenVMS AXP (Client), V2.2B	6.2	Yes	Yes	No
Reliable Transaction Router for OpenVMS AXP (Client), V2.2C	6.2	Yes	Yes	No
Reliable Transaction Router for OpenVMS AXP (Full Function), V2.2B	6.2	Yes	Yes	No
Reliable Transaction Router for OpenVMS AXP (Full Function), V2.2C	6.2	Yes	Yes	No
RMS Journaling for OpenVMS AXP, V6.1	6.1	N/A	N/A	N/A
RMS Journaling for OpenVMS AXP, V6.2	6.2	N/A	N/A	N/A
SSU (Session Support Utility) for OpenVMS Systems, V2.0	6.1	Yes	Yes	Yes
Storage Library System (SLS) for OpenVMS AXP (ACS), V2.4A	6.1	Yes	No	No
Storage Library System (SLS) for OpenVMS AXP (ACS), V2.5B	6.2	Yes	No	No
Storage Library System (SLS) for OpenVMS AXP (Development), V2.4A	6.1	Yes	No	No
Storage Library System (SLS) for OpenVMS AXP (Development), V2.5B	6.2	Yes	No	No
Storage Library System (SLS) for OpenVMS AXP (Remote), V2.4A	6.1	Yes	No	No
Storage Library System (SLS) for OpenVMS AXP (Remote), V2.5B	6.2	Yes	No	No
StorageWorks Desktop Backup/Archiver for OpenVMS AXP, V2.0	6.1	Yes	No	No
StorageWorks Desktop Backup/Archiver for OpenVMS AXP, V2.0B	6.2	Yes	No	No
StorageWorks RAID Software for OpenVMS, V2.0	6.1			
StorageWorks RAID Software for OpenVMS, V2.1	6.1			
StorageWorks RAID Software for OpenVMS, V2.2	6.2	N/A	N/A	N/A
TeamRoute for ALL–IN–1 OpenVMS AXP, V1.3	6.2			
Terminal Server Manager, V2.1	6.1			
VMScluster Software for OpenVMS AXP, V6.1	6.1			
Volume Shadowing for OpenVMS AXP, V6.1	6.1	N/A	N/A	N/A
Volume Shadowing for OpenVMS AXP, V6.2	6.2	N/A	N/A	N/A
X.25 for OpenVMS AXP Systems, V1.0	6.2			
X.25 for OpenVMS AXP Systems, V1.0A	6.2	No	Yes	Yes

Table D–3 Digital Layered Products Compatible with OpenVMS VAX Version 6.2 After May 1995

Product and Version	Software Library Availability	DECnet Support		
		Ph. IV	Syn	Full
ALL–IN–1 Integrated Office System Server for VMS (MUPA), V3.2	Dec 95–May 96	N/A	Yes	No
ALL–IN–1 Office Server Options for OpenVMS AXP and OpenVMS VAX, V3.2	Dec 95–May 96	N/A	Yes	No
DEC C for OpenVMS VAX Systems, V5.1	Dec 95–May 96	N/A	N/A	N/A
DEC Pascal for OpenVMS, V5.4	Sep 95–Nov 95	Yes	Yes	Yes
DEC TCP/IP Services for OpenVMS, V4.0	Sep 95–Nov 95	N/A	N/A	N/A
DEC VTX Version OpenVMS VAX, V7.0	Dec 95–May 96	Yes	No	No
DEC/EDI for OpenVMS VAX, V2.1A	Sep 95–Nov 95	Yes	Yes	No
DECnet/OSI Version for OpenVMS VAX, V7.0	Dec 95–May 96	N/A	Yes	Yes
DECwrite for OpenVMS VAX, V3.1	Sep 95–Nov 95	N/A	N/A	N/A
DSM Version for OpenVMS VAX Systems, V7.0	Sep 95–Nov 95	Yes	No	No
NAS Base Server 200 for OpenVMS VAX Systems, V7.0	Sep 95–Nov 95	N/A	N/A	N/A
NAS Client 150 for OpenVMS VAX Workstations, V7.0	Sep 95–Nov 95	N/A	N/A	N/A
NAS Client 250 for OpenVMS VAX Workstations, V7.0	Sep 95–Nov 95	N/A	N/A	N/A
NAS Production Server 400 for OpenVMS VAX Systems, V7.0	Sep 95–Nov 95	N/A	N/A	N/A
NAS Server 300 for OpenVMS VAX Systems, V7.0	Sep 95–Nov 95	N/A	N/A	N/A
OpenDATA Manager for OpenVMS, V1.2	Sep 95–Nov 95	No	No	No
PATHWORKS for VMS, V5.0D	Dec 95–May 96	Yes	Yes	
PATHWORKS for VMS, V6.0	Jan 97–Apr 97	Yes	Yes	
POLYCENTER Capacity Planner, V3.0A	Sep 95–Nov 95	Yes		
POLYCENTER Saveset Manager for OpenVMS, V1.2	Dec 95–May 96	Yes	No	No
POLYCENTER Scheduler OpenVMS, V3.0	Dec 95–May 96	Yes	Yes	No
POLYCENTER Scheduler OpenVMS (Agent), V3.0	Dec 95–May 96	Yes	Yes	No
POLYCENTER Scheduler OpenVMS (Client), V3.0	Dec 95–May 96	Yes	Yes	No
POLYCENTER Scheduler OpenVMS (Server), V3.0	Dec 95–May 96	Yes	Yes	No
Reliable Transaction Router for OpenVMS VAX (Client), V3.2	Sep 95–Nov 95	Yes	Yes	No
Reliable Transaction Router for OpenVMS VAX (Server), V3.2	Sep 95–Nov 95	Yes	Yes	No
StorageWorks Desktop Backup/Archiver Backup Server, V2.1	Sep 95– Nov 95	Yes	No	No
VAX COBOL for OpenVMS VAX, V5.3	Sep 95–Nov 95	Yes	Yes	Yes

Table D–4 Digital Layered Products Compatible with OpenVMS Alpha Version 6.2 After May 1995

Product and Version	Software Library Availability	DECnet Support		
		Ph. IV	Syn	Full
ACMS/Japanese for OpenVMS, V4.0	Sep 95–Nov 95			
DEC C for OpenVMS AXP Systems, V5.1	Dec 95–May 96	Yes	Yes	Yes
DEC COBOL for OpenVMS Alpha Systems, V2.3	Dec 95–May 96	Yes	No	No
DEC Data Distributor Version for OpenVMS AXP, V6.1	Sep 95–Nov 95	Yes	Yes	No
DEC GKS for OpenVMS AXP Systems (Development), V6.1	Sep 95–Nov 95	Yes	Yes	Yes
DEC GKS for OpenVMS AXP Systems (Run-Time), V6.1	Sep 95–Nov 95	Yes	Yes	Yes
DEC Optical Storage Desktop Software (OSDS) for OpenVMS AXP, V3.3	Sep 95–Nov 95	N/A	N/A	N/A
DEC Optical Storage Management Software (OSMS) for OpenVMS AXP, V3.3	Sep 95–Nov 95	N/A	N/A	N/A
DEC Pascal for OpenVMS AXP Systems, V5.4	Sep 95–Nov 95	Yes	Yes	Yes
DEC PHIGS for OpenVMS AXP Systems (Development), V3.2	Sep 95–Nov 95	Yes	Yes	Yes
DEC PHIGS for OpenVMS AXP Systems (Run-Time), V3.2	Sep 95–Nov 95	Yes	Yes	Yes
DEC/EDI for OpenVMS AXP, V2.1A	Sep 95–Nov 95	Yes	Yes	No
DECnet for OpenVMS AXP, V6.3	Sep 95–Nov 95	N/A	Yes	Yes
DECnet/OSI for OpenVMS Alpha AXP, V6.3	Sep 95–Nov 95	N/A	Yes	Yes
DECprint Supervisor for XEROX for OpenVMS AXP, V1.0	Sep 95–Nov 95			
DECwrite for OpenVMS AXP, V3.1	Sep 95–Nov 95	N/A	N/A	N/A
Digital Cartridge Server Component for VMS, V3.0	Sep 95–Nov 95			
Digital NAS Base Server 200 for OpenVMS AXP Systems, V7.0	Sep 95–Nov 95	N/A	N/A	N/A
Digital NAS Client 150 for OpenVMS AXP Systems, V7.0	Sep 95–Nov 95	N/A	N/A	N/A
Digital NAS Client 250 for OpenVMS AXP Workstations, V7.0	Sep 95–Nov 95	N/A	N/A	N/A
Digital NAS Production Server 400 for OpenVMS AXP Systems, V7.0	Sep 95–Nov 95	N/A	N/A	N/A
Digital NAS Server 300 for OpenVMS AXP Systems, V7.0	Sep 95–Nov 95	N/A	N/A	N/A
ObjectBroker for OpenVMS AXP (Development), V2.5B	Sep 95–Nov 95	Yes	No	Yes
ObjectBroker for OpenVMS AXP (Run-Time), V2.5B	Sep 95–Nov 95	Yes	No	Yes
POLYCENTER Archive/Backup for OpenVMS, V1.0	Sep 95–Nov 95	Yes	No	No
POLYCENTER Console Manager for OpenVMS, V1.6	Sep 95–Nov 95	N/A	N/A	N/A
POLYCENTER Security Compliance Manager for OpenVMS, V3.0	Sep 95–Nov 95	Yes	Yes	Yes
Reliable Transaction Router for OpenVMS AXP (Client), V3.2	Sep 95–Nov 95	Yes	Yes	No
Reliable Transaction Router for OpenVMS AXP (Full Function), V3.2	Sep 95–Nov 95	Yes	Yes	No
Storage Library System (SLS) for OpenVMS AXP (Development), V1.0	Sep 95–Nov 95	Yes	No	No
StorageWorks Array Controller Facility for OpenVMS, V1.0	Sep 95–Nov 95	N/A	N/A	N/A

Index

A

B

C

D

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

X