OpenVMS AXP Version T6.1-FT4 Release Notes

Order Number: AA-Q1Z6B-TE

February 1994

This document describes features, changes, and restrictions that apply to the OpenVMS AXP Version T6.1-FT4 operating system.

Revision/Update Information:	This is a new manual.
Software Version:	OpenVMS AXP Version T6.1-FT4

Digital Equipment Corporation
Maynard, Massachusetts

February 1994

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

The postpaid Reader's Comments forms at the end of this document request your critical evaluation to assist in preparing future documentation.

ZK6352

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
- A completed Reader's Comments form (postage paid, if mailed in the United States), or a letter, via the postal service. Two Reader's Comments forms are located at the back of each printed OpenVMS manual. Please send letters and forms to:

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

You may also use an online questionnaire to give us feedback. Print or edit the online file SYS$HELP:OPENVMSDOC_SURVEY.TXT. Send the completed online file by electronic mail to our Internet address, or send the completed hardcopy survey by fax or through the postal service.

Thank you.

Contents

2 General User-Level Release Notes

3 System Management Release Notes

4 Programming Release Notes

5 Documentation Release Notes

Index

Figures

Tables

Preface

The OpenVMS AXP Version T6.1-FT4 Release Notes manual describes features and restrictions of the OpenVMS AXP Version T6.1-FT4 operating system. It also provides information about operating AXP computers and provides descriptions of features that are unique to AXP systems.

Intended Audience

This manual is intended for general users, system managers, developers, and programmers who use the operating system. It is especially useful for those who are about to install the operating system.

Document Structure

Chapter 1 provides information about using AXP computers.

Chapter 2 describes changes, new features, and restrictions that affect some commonly used procedures and utilities.

Chapter 3 describes changes, new features, and restrictions that affect utilities used for system and performance management, system maintenance, and networking.

Chapter 4 provides information about changes, new features, and restrictions that affect areas such as tools, utilities, run-time libraries, system services, and the file system.

Chapter 5 describes corrections to the OpenVMS AXP documentation set.

Associated Information

For a list of additional documents that are available in support of this version of OpenVMS AXP, refer to the CD–ROM booklet and 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. The *VMS for Alpha Platforms Internals and Data Structures* manual mentioned in this document is published by Digital Press under order number EY–L466E–P1.

Conventions

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

The following conventions are also used in this manual:

V1.x
A margin note beneath each heading indicates the OpenVMS AXP version in which the note was first published. All notes identified with the *V1.0* or *V1.5* margin note also apply to OpenVMS AXP Version T6.1-FT4.

Ctrl/*x*
A sequence such as Ctrl/*x* indicates that you must hold down the key labeled Ctrl while you press another key or a pointing device button.

PF1 *x*
A sequence such as PF1 *x* indicates that you must first press and release the key labeled PF1, then press and release another key or a pointing device button.

GOLD *x*
A sequence such as GOLD *x* indicates that you must first press and release the key defined GOLD, then press and release another key. GOLD key sequences can also have a slash (/), dash (-), or underscore (_) as a delimiter in EVE commands.

Return
In examples, a key name is shown enclosed in a box to indicate that you press a key on the keyboard. (In text, a key name is not enclosed in a box.)

. . .
In examples, a horizontal ellipsis indicates one of the following possibilities:

- Additional optional arguments in a statement have been omitted.
- The preceding item or items can be repeated one or more times.
- Additional parameters, values, or other information can be entered.

.
.
.
A vertical ellipsis indicates the omission of items from a code example or command format; the items are omitted because they are not important to the topic being discussed.

( )
In format descriptions, parentheses indicate that, if you choose more than one option, you must enclose the choices in parentheses.

[ ]
In 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 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.

Boldface text is also used to show user input in Bookreader versions of the manual.

italic text
Italic text emphasizes important information, indicates variables, and indicates complete titles of manuals. Italic text also represents information that can vary in system messages (for example, Internal error *number*), command lines (for example, /PRODUCER=*name*), and command parameters in text.

UPPERCASE TEXT
Uppercase text indicates a command, the name of a routine, the name of a file, or the abbreviation for a system privilege.

Hyphens in coding examples indicate that additional arguments to the request are provided on the line that follows.

numbers

All numbers in text are assumed to be decimal, unless otherwise noted. Nondecimal radixes—binary, octal, or hexadecimal—are explicitly indicated.

mouse

The term *mouse* refers to any pointing device, such as a mouse, a puck, or a stylus.

MB1, MB2, MB3

MB1 indicates the left mouse button, MB2 indicates the middle mouse button, and MB3 indicates the right mouse button. (The buttons can be redefined by the user.)

1 Running OpenVMS AXP

This chapter discusses aspects of the AXP operating system environment, including installation and upgrade capabilities, running DECwindows software, and using AXP computers.

1.1 Release Notes Specific to OpenVMS AXP Version T6.1-FT4

V2.0 The release notes in this document are cumulative from OpenVMS AXP Version 1.0. In this chapter, the following sections contain release notes that pertain specifically to OpenVMS AXP Version T6.1-FT4:

- TBS

1.2 Compatibility with Translated Programs and Images

V1.5 Digital supports translation of programs from VAX VMS Version 4.0 through Version 5.4-3. Although the VAX Environment Software Translator (VEST) translates Version 5.5 images, OpenVMS AXP Version 1.5 might not provide the necessary run-time support because the translated libraries are based on Version 5.4-3. In this case, the translated VAX VMS Version 5.5 image may get an identification mismatch at run time.

1.3 Privileged Architecture Library Code (PALcode) Version Checking Enabled

V1.5 If your computer console does not have the specified minimum version of the Privileged Architecture Library code (PALcode), you may not be able to boot your system during the installation or upgrade procedure. Digital recommends, therefore, that you do the following before performing an installation or upgrade:

1. At the console prompt (>>>) on your running AXP system, enter the SHOW CONFIGURATION command (or SHOW PAL on DEC 7000 and DEC 10000 AXP series systems). The system display indicates which version of PALcode is running.
2. Refer to the cover letter that accompanied your kit for the absolute (required) minimum and recommended minimum versions of PALcode for your AXP computer.
3. Proceed as follows, depending on which version of PALcode your system is currently running:
   - If the PALcode is below the *absolute* (required) minimum, upgrade your console to the required minimum version (or higher) listed for your AXP computer in the cover letter that accompanied your kit. You can accomplish this task either by following directions contained in the hardware manuals that came with your computer or by contacting your Digital representative.

Caution

If you do not upgrade your console to the absolute (required) minimum PALcode, you receive a fatal error message similar to the following when you attempt to boot your system:

```
APB-F-PALREV, PALcode revision 5.15 is below required minimum of 5.25
        UNABLE TO CONTINUE
```

- If the PALcode is below the *recommended* minimum, you receive a warning message similar to the following:

  ```
  APB-W-PALREV, PALcode revision 5.25 is below recommended minimum of 5.39
  ```

  Although you are still able to boot your system, you should review the cover letter and check with your Digital representative to determine how you should upgrade your console to the recommended minimum version of the PALcode before installing or upgrading the operating system.

1.4 DEC 2000 Model 300 Notes and Restrictions

V2.0 This section contains notes and restrictions specific to DEC 2000 Model 300 systems.

1.4.1 No Datacheck Support for Floppy Drive

V2.0 The OpenVMS AXP operating system does not include support for datacheck operations for the floppy diskette drive (DVDRIVER) on DEC 2000 Model 300 computers.

1.4.2 Using the EISA Configuration Utility (ECU)

V2.0 When deciding whether to use the EISA Configuration Utility (ECU) on DEC 2000 Model 300 systems, note the following:

- You cannot run the ECU from an alternate console.
- If you want to configure an EISA device other than the base configuration, you must use the ECU.
- If you have been using the ECU on a Windows NT™ system, you must continue to do so. However, because the OpenVMS ECU configuration files are different (most significantly in the configuration of multiple Adaptec™ SCSI adapters) from those used by the Windows NT system, you should rerun the ECU specifically provided for OpenVMS or OSF systems.

  To rerun the ECU, do the following:

  1. Because you must run the ECU from the Windows NT console, enter the following command:

     ```
     SET OS NT
     ```

  2. After power cycling the system, use the arrow key to select the SETUP menu. Using this menu, you can select and run the ECU from a floppy diskette.

™ Windows NT is a trademark of Microsoft Corporation.
™ Trademark of Adaptec, Inc.

3. For complete information about using the ECU, refer to the ECU online help and the manual titled *DECpc AXP 150 and DEC 2000 Model 300 Customer Technical Information*.

Note

If you make a copy of the ECU floppy diskette, set the volume label to SYSTEMCFG.

1.4.3 Running AUTOGEN

V2.0 Digital strongly recommends that you run AUTOGEN to make sure all system parameter values are calculated correctly. (Be sure the system has been running for at least 24 hours with a typical workload before running AUTOGEN.)

1.4.4 SCSI Port Error Count

V2.0 When the OpenVMS AXP operating system is booted on a DEC 2000 Model 300 system, SCSI ports labeled PK*n*0 (as listed in the SHOW DEVICE or SHOW ERROR display) will show an error count of 1 but will not have the error listed in the log file. This is not an actual error, but rather the result of the SCSI bus reset issued in the boot path.

1.4.5 Switching Console Output

V2.0 You can switch your console output to the alternate console or to the graphics terminal by using the SET CONSOLE command and specifying either **VGA** (graphics terminal) or **serial** (alternate console). For example:

```
>>> SET CONSOLE VGA
```

1.4.6 Keyboard Support on DEC 2000 Model 300 Systems

V2.0 The DEC 2000 Model 300 keyboard comes in both 101- and 102-key versions. This keyboard looks similar to the LK443 (101-key) and LK444 (102-key) keyboards supplied with other Digital computers. This section describes modifications that make it easier for you to use the PC keyboard in an OpenVMS environment.

1.4.6.1 Keyboard Mappings

V2.0 The keyboard uses an industry-standard PC layout, rather than an LK201 or LK401 keyboard layout. However, to be compatible with Digital software, the default keyboard mapping on DEC 2000 Model 300 systems does not always generate the value that is engraved on the key, but instead generates the value that would be applicable to an LK201 or LK401 keyboard. This allows most OpenVMS applications to work using the default keyboard without requiring you to use shift keys for commonly used keys.

1.4.6.2 Using the CAPS LOCK Key

V2.0 By using the CAPS LOCK key as a shift key and then pressing function keys F1 through F10 and the Keypad Plus Key, you can generate function keys F11 through F20 and the Keypad Minus key. When any of these keys are pressed, and the CAPS LOCK key is pressed, the CAPS LOCK is assumed to be a temporary shift key and the LOCK state is returned to its previous state, and the translated function key is generated. This provides compatibility with the VT510 terminal and allows the generation of keys F17 to F20 by pressing CAPS LOCK and then keys F7 to F10.

1.4.6.3 Keyboard Restrictions

V2.0

Note the following keyboard restrictions:

- This release does not support Num Lock as a locking modifier for the keypad, as it does on a PC. Num Lock support will be provided in a future release of the OpenVMS AXP operating system.
- The Num Lock light corresponds to the COMPOSE light.
- You can activate the COMPOSE function by using the LK201 mechanism of pressing the ALT and SPACE keys.
- This release does not set the default keyboard type and language and will instead use a LK401 keymap file. To obtain the correct keyboard mapping, you must select the Keyboard entry from the Options pulldown menu in the Session Manager window. The keymap files are denoted by LK444 (102-key keyboards) and LK443 (101-key US keyboard).

1.4.7 Graphics Support on DEC 2000 Model 300 Systems

V2.0

The following notes and restrictions apply to DEC 2000 Model 300 systems with graphics.

- For OpenVMS AXP Version T6.1-FT4 field test, the system parameter VIRTUALPAGECNT must be set to a minimum of 500,000 pagelets. The server requires over 160MB of virtual address space to map the frame buffer and registers into the DEC 2000 Model 300 sparse address space.
- The DEC 2000 Model 300 computer supports only a single QVision video card.
- Graphics will not work if you boot the system using the alternate console. Only the console for graphics (with keyboard and mouse input) is supported.
- When you boot the system from the graphics console, the serial lines do not work, nor can you use Ctrl/P to halt the system. (Although Ctrl/P will halt the system, the keyboard will not be left in the correct state. Use the Halt button instead.)
- If the window system is active, halting the system and then entering the console command, CONTINUE, is not supported.
- Note the following about the console window:
  - The console window is full screen (VGA mode) and when active, the server operation is suspended until the console window is removed.
  - Because the server is stalled by the console, a logical is available that will allow you to remove the console window automatically after a specified number of seconds.

    This feature is not enabled by default. To enable the feature, define the server logical (in DECW$PRIVATE_SERVER_SETUP.COM) as follows:

    ```
    DEFINE/SYS DECW$SERVER_AUTO_CONSOLE_REMOVE n
    ```

    where:

    n is the number of seconds you want the console window to be visible before it is removed.

- The console window is 24 lines by 80 columns. The 25th line contains one of two possible displays. When the window system is active, the following message is displayed:

```
OpenVMS AXP Operator Console - Ctrl+F2 to Resume the Window System
```

When the window system is not active, the following message is displayed:

```
OpenVMS AXP Operator Console
```

You can customize or internationalize the messages by defining the server logicals (in DECW$PRIVATE_SERVER_SETUP.COM) as follows:

```
DEFINE/SYS DECW$SERVER_ACTIVE_STATUS_LINE "text-for-active-window-system-message"
DEFINE/SYS DECW$SERVER_NOT_ACTIVE_STATUS_LINE "text-for-inactive-window-system-message"
```

- The server implements a fast 0-width line drawing algorithm that uses the QVision drawing engine at full speed. Due to a hardware problem, lines may not be correct for the last pixel of lines using CapNotLast. You can choose accurate semantics but the drawing performance will be 10 times slower (0-width lines do not need to be accurate). To draw end pixels correctly for 0-width lines, define the server logical (in DECW$PRIVATE_SERVER_SETUP.COM) as follows:

```
DEFINE/SYS DECW$SERVER_USE_SLOW_VGA TRUE
```

1.5 Running OpenVMS AXP Using a DEC 3000 Series AXP Computer

V1.0 The following sections discuss features and configuration tasks that are common to the DEC 3000 series AXP computers.

1.5.1 Running DECwindows

V2.0 Support for the DECwindows Motif for OpenVMS AXP layered product is provided by:

- The DECwindows X11 display server, which is an optional part of the AXP kit. This optional part can be selected during the AXP installation procedure, when you are asked if you want to provide optional DECwindows workstation support.
- The DECwindows Motif for OpenVMS AXP layered product, which provides programming and application support. This layered product must be installed separately.

To run DECwindows Motif for OpenVMS AXP applications locally on the DEC 3000 series AXP computers, you must select the option for the DECwindows X11 display server during the AXP installation procedure and also install the DECwindows Motif for OpenVMS AXP layered product. See the DECwindows Motif for OpenVMS AXP Version 1.1 Installation Guide for information about installing the DECwindows Motif for OpenVMS AXP Version 1.1 layered product.

1.5.2 DECwindows Support for OpenVMS AXP

Note the following about required and optional DECwindows save sets:

- The required DECwindows save set included with the operating system supplies the base support you need to run the DECwindows Motif for OpenVMS AXP layered product, which you install separately.
- The optional DECwindows workstation support save set that you can choose during the AXP installation is required if you want to run the windowing software on a DEC 3000 series AXP workstation (provided you also install the separate DECwindows Motif for OpenVMS AXP layered product).
- The required DECwindows save set plus the optional DECwindows workstation support save set offered with the operating system does *not* constitute complete DECwindows support. You *must* install the separate DECwindows Motif for OpenVMS AXP layered product to obtain complete support.

Note the following as well:

- The DECwindows Motif for OpenVMS AXP layered product supports both the Motif *and* XUI environments.
- If you want to customize aspects of the DECwindows environment, see the appendix in the *DECwindows Motif Version 1.2 for OpenVMS Installation Guide* for instructions.

1.5.3 Using DECW$TAILOR to Add or Remove DECwindows Files

For OpenVMS AXP, you can use DECW$TAILOR to add or remove some or all DECwindows files from your system disk.

1.5.4 DECwindows X11 Display Server and Driver

The DECwindows X11 display server, including device drivers and fonts, can be selected for installation during the AXP installation procedure. The display server implementation is based upon the MIT X Window System Version 11 Release 5 (R5) display server.

The following display server extensions are supported and have been tested:

- Adobe-DPS-Extension — Display PostScript extension
- DEC-XTRAP — X protocol trap extension
- MIT-SUNDRY-NONSTANDARD — MIT miscellaneous extension
- Multi-Buffering — Multibuffering extension
- SHAPE — Nonrectangular windows extension
- XIE — X imaging extension

1.5.4.1 VIRTUALPAGECNT Requirements

V2.0

When using the HX graphics adapter, the system parameter VIRTUALPAGECNT must be set to a minimum of 150000 pagelets for a workstation with a single graphics adapter. For each additional graphics adapter, VIRTUALPAGECNT should be increased by 100000 pagelets.

If VIRTUALPAGECNT is not sufficient the display server will not run.

1.5.4.2 Loadable Server Extensions

V2.0 The OpenVMS display servers support dynamically loaded extensions. These extensions are contained in separate shareable images and are activated on demand.

The display server requires that certain dynamically loaded extensions to be loaded at initialization time. It uses the symbol DECW$SERVER_EXTENSIONS to identify which extensions to automatically load. By default the following extensions are loaded:

- Adobe-DPS-Extension
- Xie
- DEC-XTRAP
- Multi-buffering

If you do not want to use system resources to support any of these extensions add the DECW$SERVER_EXTENSIONS symbol to the site-specific server setup file SYS$MANAGER:DECW$PRIVATE_SERVER_SETUP.COM. For example, you can exclude the Display Postscript extension by defining the symbol as follows:

```
DECW$SERVER_EXTENSIONS == "Xie,DEC-XTRAP,Multi-buffering"
```

Note: Do not leave any white space in the quoted string.

1.5.4.3 New Fonts

The Lucida family of fonts were added for OpenVMS AXP Version T6.1-FT4. It is available in 8, 10, 12, 14, 18, 19, and 24 point sizes in a number of different font styles.

A new VT330 font family for DECterm was added for OpenVMS AXP Version T6.1-FT4.

The DEC VT330 font offers full DEC VT emulation and is design-consistent with the font used in Digital's Video Terminal products. It provides the full ISO8859-1 (ISO Latin-1) and DEC Technical character sets.

This font is the same font as that built into Digital's popular VT330 and VT340 series video terminals. Like its hardware counterparts, the VT330 font provides a base character cell of 10 by 20 pixels and is designed for use with monitors that display 100 dots/in. The font is "pixel-for-pixel" compatible with the VT330, VT340, and VT340+ devices. Twenty-four files comprise the full DEC VT330 font set. The DEC VT330 font currently ships with the VXT2000 InfoServer software products.

The DEC VT330 font is designed specifically for use with the DECterm terminal emulator in an X Window environment and provides an alternative to the standard DECterm "Big" font, based on Bitstream Courier. It provides a DECterm font that fits between the "Little" and "Big" fonts, provided with every DECwindows implementation. It also allows ReGIS Graphics applications to more accurately position text characters amid graphics. The DEC VT330 font helps satisfy both of these concerns. In addition to being stylistically distinctive, DEC VT330 provides better ReGIS emulation than any of the standard DECterm ("terminal") fonts because its character-cell aspect ratio is exactly 2-to-1. At 100 dots/in, DEC VT330 is a 14 point font. Used with 75 dots/in displays, DEC VT330 is an 18 point font.

DECterm can access the DEC VT330 font by specifying the following XLFD font name string as its bigFontSetName resource:

-Dec-Vt330-*-*-*-20-*-*-*-C-*-*-*

1.5.4.4 New Font Formats

The display server supports the portable compiled format (PCF) that was introduced with the MIT R5 display server. PCF is a modified font format that includes bit/byte ordering and alignment information as part of the font data and allows for vendor-independent font support. All fonts currently provided with the DECwindows display server are available in PCF format. Applications do not have to be modified in order to use the PCF fonts.

1.5.4.5 Font Compiler — Restrictions

The DECwindows X11 display server includes a font compiler that enables programmers to convert an ASCII bitmap distribution format (BDF) into binary portable compiled format (PCF). The server uses a PCF file to display a font. Note that the DECwindows X11 display server for the OpenVMS VAX font compiler's MINBBOX and REPORT qualifiers are not supported in OpenVMS AXP.

To invoke the font compiler, use the following format:

FONT filespec [/OUTPUT[=*filename*] [/[NO]INK_METRICS]

The *filespec* parameter specifies the BDF file to be converted to PCF format. A file name is required. The default value of the optional file type is .BDF.

The /OUTPUT qualifier specifies the file name and file type of the resulting PCF file. The default output file name is the file name of the BDF file being converted. The default output PCF file type is .PCF.

PCF output files consist of a header file that contains font information, character metrics, amd the image of each character in the font.

The /NOINK_METRICS qualifier causes the font compiler to omit the font's ink metrics information (per character). This will reduce overall font in-memory usage. The /INK_METRICS qualifier is used by default. Ink metrics define the smallest box that includes the entire symbol, without any extraneous space outside the symbol.

1.5.4.6 Font Server Support

The R5 display server includes a font server renderer that supports the use of font servers to access fonts. This allows font files to be distributed on systems other than the system on which the display server is running.

You can add font servers to the font path by defining the symbol DECW$FONT_SERVERS in the site-specific server setup file SYS$MANAGER:DECW$PRIVATE_SERVER_SETUP.COM. Depending on the transport type, use either of the following:

- To add support for the TCP transport, specify TCP as the network connection type that will communicate with the font server. Replace *node* with the Internet node name. Specify the TCP/IP port number for *port_number*, as shown in the following example:

  ```
  $ DECW$FONT_SERVERS == "TCP/node::port_number"
  ```

- To add support for the DECnet transport, specify DECNET as the network connection type that will communicate with the font server. Replace *node* with the DECnet node name. Specify the DECnet object name for *network_object*.

  ```
  $ DECW$FONT_SERVERS == "DECNET/node::network_object"
  ```

Multiple font servers can be added by defining the symbol as a comma-separated list. The symbol is not case sensitive.

1.5.4.7 Support Added for Scalable Outline Fonts

The DECwindows server supports scalable outline fonts. You can select an arbitrary size for any of these fonts, and they will be scaled as necessary; otherwise, they are treated as standard bitmapped fonts. The following scalable fonts are included:

- Adobe-Courier-Bold-I-Normal–0-0-0-0-P-0-ISO8859-1
- Adobe-Courier-Bold-R-Normal–0-0-0-0-P-0-ISO8859-1
- Adobe-Courier-Medium-I-Normal–0-0-0-0-P-0-ISO8859-1
- Adobe-Courier-Medium-R-Normal–0-0-0-0-P-0-ISO8859-1
- Adobe-Utopia-Bold-I-Normal–0-0-0-0-P-0-ISO8859-1
- Adobe-Utopia-Bold-R-Normal–0-0-0-0-P-0-ISO8859-1
- Adobe-Utopia-Medium-I-Normal–0-0-0-0-P-0-ISO8859-1
- Adobe-Utopia-Medium-R-Normal–0-0-0-0-P-0-ISO8859-1
- Bitstream-Charter-Bold-I-Normal–0-0-0-0-P-0-ISO8859-1
- Bitstream-Charter-Bold-R-Normal–0-0-0-0-P-0-ISO8859-1
- Bitstream-Charter-Medium-I-Normal–0-0-0-0-P-0-ISO8859-1
- Bitstream-Charter-Medium-R-Normal–0-0-0-0-P-0-ISO8859-1
- Bitstream-Courier-Bold-I-Normal–0-0-0-0-M-0-ISO8859-1
- Bitstream-Courier-Bold-R-Normal–0-0-0-0-M-0-ISO8859-1
- Bitstream-Courier-Medium-I-Normal–0-0-0-0-M-0-ISO8859-1
- Bitstream-Courier-Medium-R-Normal–0-0-0-0-M-0-ISO8859-1
- Bitstream-Courier-Medium-R-Normal–0-0-0-0-M-0-ISO8859-1

Font names that contain zeros in the size fields, such as those listed above, indicate scalable fonts.

When selecting any of the fonts listed above, supply either the pixel size (the first zero in the size field) or the decipoint size (the second zero in the size field) to obtain the desired font size.

For example, the following font specifications both select a 12-point Courier font:

```
                          Size in pixels
                                |
Adobe-Courier-Medium-R-Normal--17-*-*-*-p-*-ISO8859-1

Adobe-Courier-Medium-R-Normal--*-120-*-*-p-*-ISO8859-1
                                  |
                         Size in decipoints
```

The pixel size of a 12-point font at 100 dots per inch is 17 (120*100/720). If you specify values for both the pixel size and the decipoint size and those values are incompatible, no font is selected. If you omit both the pixel size and the decipoint size, a default size of 12-point is selected.

The private display server setup file SYS$MANAGER:DECW$PRIVATE_SERVER_SETUP.TEMPLATE contains information about how to customize a private display server command procedure for your environment. Rename the SYS$MANAGER:DECW$PRIVATE_SERVER_SETUP.TEMPLATE to a command procedure file type (.COM). Edit the resulting SYS$MANAGER:DECW$PRIVATE_SERVER_SETUP.COM command procedure, and define the symbol DECW$MONITOR_DENSITY to be a value that represents the desired screen resolution. The symbol's value is rounded to either 75 or 100 dots per inch and is used to fill in the remaining fields in the font name. The default value is 100 dots per inch.

1.5.4.8 Keyboard support

New keyboard keymap files were added for OpenVMS AXP Version T6.1-FT4. The file type is .DECW$KEYMAP. These keymap files support LK444, LK443 and PCXAL keyboards for the various countries listed below. Reference the Keyboard support on DEC 2000 Model 300 systems for more information about these keyboards.

- BELGIAN_LK444_VT
- DENMARK_LK444_VT
- FRANCE_LK444_VT
- GERMANY_LK444_VT
- ITALY_LK444_VT
- NORWAY_LK444_VT
- PORTUGAL_LK444_VT
- SPAIN_LK444_VT
- SWEDEN_LK444_VT
- SWISS_LK444_VT
- UK_LK444_VT
- US_LK443_VT

1.5.4.9 DECwindows LAT support

V2.0 DECwindows now supports LAT as a network transport mechanism for displaying to OpenVMS Alpha AXP workstations.

Before LAT transport can be used, the LAT software must be started on both the DECwindows client and server systems. Refer to the LATCP Utility Reference Manual for details on starting and configuring the LAT software.

In order to use LAT as a transport the LAT service called "X$SERVER" must be present on the display server system. Define the following logical to force the DECwindows startup procedure to create the LAT service.

```
$ DEFINE /SYSTEM DECW$INSTALL_XTERMINAL SERVER
```

This logical must be defined before DECwindows is started. It is suggested that you define this logical in SYS$MANAGER:SYLOGICALS.COM.

To enable LAT as a transport for your display server, modify the file SYS$MANAGER:DECW$PRIVATE_SERVER_SETUP.COM to include the LAT transport in the list of possible transports. For example:

```
$ decw$server_transports == "LOCAL, DECNET, TCPIP, LAT"
```

If there is no SYS$MANAGER:DECW$PRIVATE_SERVER_SETUP.COM on the system, create one by copy the file SYS$MANAGER:DECW$PRIVATE_SERVER_SETUP.TEMPLATE to SYS$MANAGER:DECW$PRIVATE_SERVER_SETUP.COM.

After modifying the server setup file, restart the DECwindows server:

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

When you want to run an application using LAT, specify LAT as the transport type when entering the DCL command to set the display.

```
$ set display/create/node=nname/transport=LAT
```

1.5.4.10 DECwindows Transport

V2.0 Note: This information was also documented in the V1.5 release and included here for those who may not have made these changes at the time of that release. If you have made the changes outlined below, all that is required for OpenVMS AXP Version T6.1-FT4 is a recomplile and relink of the user-written transport.

The DECwindows transport has been modified to correct disconnect problems. Under certain conditions, pending asynchronous system traps (ASTs) were sent after the transport cleanup process had begun. As a result, the AST attempted to access data that had already been deallocated, which in turn caused access violations or system failures.

One solution, queuing the CLOSE_AND_DEALLOCATE_AST AST from the DECW$$TCPIP_CLOSE routine, was an incomplete one. The CLOSE_AND_DEALLOCATE_AST AST was queued to the current operation mode (executive), 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 was increased by 1 longword to provide an additional routine entry, XTFT$A_DISCONNECT. The new common routine is DECW$XPORT_DISCONNECT.

Digital recommends that you implement the following modifications if you have written your own DECwindows transport. These modifications are based on the sample transport implementation 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$ADISCONNECT. The XTFT is initialized in the DECW$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$XPORT_DISCONNECT, passing an IXTCC parameter. The DECW$XPORT_DISCONNECT routine calls the routine stored in XTFT[XTFT$A_DISCONNECT]. By dispatching the disconnect routine

through the Common Transport layer, the process changes to executive mode as required to deallocate data structures.

Restrictions

The following restrictions apply to the DECwindows transport:

- Although the VMS DECwindows Transport Manual does not state it, routines in customer-supplied DECwindows transports must execute in executive mode.

  If you are writing your own transport layer, the transport-specific routines must be executable in executive mode. This restriction applies to both AXP and VAX DECwindows transport routines.
- DECwindows DECnet, TCP/IP, LAT, and local transport display server options are available.
- Even though your application disables user-mode ASTs before calling XLIB, you may still receive user-mode ASTs in some circumstances. (This is a problem with XLIB.)

1.5.4.11 Alternate Window System Console Provides 24-Line Console Window

Normally, when you press the halt button and continue system operation, only six lines are displayed by the console at the top of the screen. This action minimizes console offscreen memory usage and allows maximum offscreen memory usage by the display server for pixmap and font glyph storage. This is a permanent restriction in the display server. When you turn on or reboot the system, the full console screen is displayed.

However, an alternate window system console (also called a software console) is available. The software console provides a 24-line console window when the window system is running instead of the current 6-line window that is available through the firmware callbacks.

Note that the software console destroys the display and does not use offscreen memory for its display area. As a result, when the software console window is removed, the server generates exposure events that redraw the display on the windows that were obscured by the console. The main benefit of exposure event generation is that approximately 153,600 bytes of offscreen memory are available to the server.

However, the software console has the following limitations:

- Windows with save unders (for example, Motif pulldown menus) are not correctly exposed when the console is removed.

  To work around this problem, either disable save unders or select the Refresh Window item in the Menus... option.
- Input focus in the software console window might cause the Workspace menu to corrupt the display.
- With the Motif window manager, new window creation can sometimes create the window over the software console window. (This does not happen with the XUI window manager.)

Usually, you can correct these problems by pressing Ctrl/F2 twice to redraw the console area.

To select the alternate window system console, set bit 8 (value 512) in the TTY_DEFPORT system parameter. Digital recommends that you use the System Management utility (SYSMAN) to modify system parameters. You can also set the TTY_DEFPORT system parameter during an interactive boot by using the command SET TTY_DEFPORT 512.

1.5.4.12 Known Problems

- Oscillator selection on nonconsole monitors may be improperly initialized if the system is configured with multiple PMAGB-BA (HX) options and if the nonconsole monitors do not use the default oscillator. The VRT19-DA is one example of a monitor that does not use the default oscillator. The symptoms are a compressed screen with text shifted to the left and no flashing cursor on the lower left edge of the screen.

  Work around this problem by editing the SYS$MANAGER:SYSTARTUP_VMS.COM file and defining the following multivalued logicals:

  ```
  $ DEFINE/SYS/EXEC DECW$SERVER_MONITOR_RESET  0,1[,...]
  $ DEFINE/SYS/EXEC DECW$SERVER_MONITOR_SELECT 0,1[,...]
  ```

  Each value in the logical applies to a specific head (starting from screen 0). Specify a value for each head. Any value not provided is taken from the initial value.

  When set to a logical TRUE value, the RESET logical tells the X11 display server to unconditionally select the HX oscillator. A logical FALSE value for the RESET logical prevents the display server from altering the oscillator. A logical TRUE value for the SELECT logical selects the alternate oscillator (used by the VRT19-DA) and a logical FALSE value selects the standard oscillator (used by the VR320).

  This problem is caused by a firmware problem that Digital expects to fix in a future release.

- Console messages can corrupt DECterm windows with vertical bars using PMAGB-BA (HX) Turbo Channel graphics device support. To work aroound this problem, select the Refresh Window item in the Session Manager's Menus... option.

1.5.4.13 Error Recovery

The display server contains a condition handler that intercepts problems (such as an access violation) that might otherwise cause the display server to stop. The condition handler tries to let the display server continue. When the condition handler intercepts a problem, it sends an implementation error to the client or disconnects the client or both.

When the condition handler recovers from an error, the display server might lose resources, such as memory. Therefore, after a number (10) of these exceptions, the condition handler broadcasts a warning message to all users on the workstation indicating that the display server might be running in a degraded mode and suggesting that the display server be restarted. If you see messages like this, you should restart the display server at the next convenient opportunity. Enter the following command from a privileged account to restart the display server:

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

1.5.4.14 Reporting Errors

The file SYS$MANAGER:DECW$SERVER_0_ERROR.LOG contains display server startup and operational error information. Use the contents of this file when you submit problem reports concerning the display server.

1.5.5 Multihead Configurations

A multihead configuration consists of a single DEC 3000 series AXP workstation that supports multiple graphics options. A graphics option consists of a graphics controller and a graphics display interface (monitor).

A DEC 3000 series AXP system will be automatically configured for multihead use if you rename the private display server setup file from a template file type (.TEMPLATE) to a command procedure file type (.COM). Rename this file after installing the DECwindows X11 display server and after logging on to your system.

To rename the private display server setup file, enter the following command:

```
$ RENAME SYS$MANAGER:DECW$PRIVATE_SERVER_SETUP.TEMPLATE
_To: SYS$MANAGER:DECW$PRIVATE_SERVER_SETUP.COM
```

The display server loads the private display server setup command procedure on startup or restart.

The private display server setup file always configures the console as the primary head (screen 0). Note that the firmware always selects the device with the lowest TURBOchannel slot address found in the system as the console device.

After you have renamed the private display server setup file, use the following command to restart the display server:

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

1.5.6 TURBOchannel PMAZB Adapter Support

V1.5 OpenVMS AXP Version 1.5 contains support for the PMAZB TURBOchannel adapter. The PMAZB adapter is software-compatible with the integrated Small Computer System Interface (SCSI) ports on the DEC 3000 series AXP systems.

The DEC 3000 series AXP system consoles implement the SHOW CONFIGURATION console command, which displays information about the TURBOchannel options and the built-in adapters in the system. When a PMAZB adapter is installed in the TURBOchannel, the SHOW CONFIGURATION command displays the "PMAZB-AA" string, the TURBOchannel slot number, and the device status.

The DEC 3000 series AXP consoles also implement the SHOW DEVICE command, which displays information about the devices in the system. Because the integrated SCSI adapter is built into every DEC 3000 series AXP system, the SHOW DEVICE console command can display the SCSI devices connected to the integrated SCSI ports. However, the SHOW DEVICE console command cannot display the SCSI devices connected to the PMAZB SCSI ports.

To make the console display the devices connected to the PMAZB SCSI ports, enter the following command at the console prompt, where *n* is the TURBOchannel slot number in which the PMAZB adapter is installed:

```
>>> test tcn cnfg
```

This command displays the devices that are connected to each SCSI port of the PMAZB adapter. The device controller letters are either A or B, based on the PMAZB ports to which the devices are connected. Do not confuse these devices with any DKA*xxx* or DKB*xxx* devices displayed by the SHOW DEVICE command, which shows SCSI devices on the integrated SCSI ports only.

See the *OpenVMS AXP Version 1.5 Upgrade and Installation Manual* for information about how to boot from a device connected to a PMAZB adapter.

OpenVMS AXP does not distinguish between the PMAZB adapter and the integrated SCSI adapter. The operating system views them as identical adapters. Because OpenVMS AXP searches for I/O adapters in backplane slot number order, device controller letters are assigned that correspond to the backplane order of the TURBOchannel options, followed by the integrated adapters. This is different from console SCSI device naming, which always designates SCSI devices on the integrated SCSI ports as either "A" or "B" port devices.

For example, on a DEC 3000 Model 500 AXP system with no TURBOchannel options installed, OpenVMS AXP names the integrated SCSI ports PKA0 and PKB0, and the devices connected to the ports inherit the controller letter from the port controller letter (A or B). In this case, the OpenVMS AXP SCSI device naming matches the console SCSI device naming. However, if a PMAZB adapter is installed in the TURBOchannel, OpenVMS AXP names the PMAZB SCSI ports PKA0 and PKB0 and names the integrated SCSI ports PKC0 and PKD0. The devices connected to the ports inherit the controller letter from the port controller letter (A, B, C, or D).

1.5.7 Displaying Model Numbers for DEC 3000 Model 600 and 800 Systems

V2.0 When you enter the SHOW CLUSTER command, the HW_TYPE field displays "ALPH" instead of specific model numbers for the DEC 3000 Model 600 and 800 systems.

For example:

```
$  SHOW CLUSTER/CONTINUE

View of Cluster from system ID 62345  node: DEC3T6        27-JUL-1993 16:49:20

                                                          SYSTEMS

  NODE    SYS_ID            HW_VERS                 HW_TYPE        SOFTWARE

 DEC3T6     62345  042700000000000000000004  ALPH                  VMS V1.5
 DEC3T8     20073  042800000000000000000004  ALPH                  VMS V1.5
 DEC3T4     64568  040B00000000000000000004  DEC 3000 Model 400    VMS V1.5
 DEC3T5     65525  040900000000000000000004  DEC 3000 Model 500    VMS V1.5
 VAX4T6     64894  00EC00000000000012000003  VAXstation 4000-60    VMS V6.0
```

To display the model number for a DEC 3000 Model 600 or 800 system, log in to that system and then enter the following command:

```
$ Write SYS$OUTPUT F$GETSYI ("HW_NAME")
```

1.6 Running OpenVMS AXP Using a DEC 4000 Series AXP Computer

V1.0 Note that the ANALYZE/ERROR command does not format error log entries resulting from DSSI devices. This problem will be corrected in a future release of the operating system.

1.7 Running OpenVMS AXP Using a DEC 7000 Model 600 AXP Computer

V1.0 The following sections discuss features and problems related to DEC 7000 Model 600 AXP Computers.

1.7.1 Local DIGITAL Storage Architecture (DSA) Disk and Tape Support

V1.0 The operating system provides support for local DSA disk and tape devices on the DEC 7000 Model 600 AXP computer. Support is provided with the KDM70 on the XMI bus. However, you must obtain the latest KDM microcode, which supports the AXP 8-kilobyte page size, from your Digital representative.

See the *OpenVMS AXP Software Product Description* (SPD 41.87.xx) for information about devices and peripherals that are supported by the operating system.

1.7.2 SCSI Support

V1.0 The host ID is 7 for ports A and B of the KZMSA-AB SCSI controller.

1.7.3 Booting from a SCSI Disk — Problem When XDELTA Is Included

V1.5 A problem in Version 1.5 results in an error halt on the DEC 7000 Model 600 AXP SMP computer when the following conditions are met:

- Your system boots from a SCSI disk.
- You issue a boot command that instructs the system to include XDELTA at boot time and to stop at the EXEC_INIT breakpoints.

  The boot command would be similar to that shown in the following example. In the example, *n* is the root from which you boot, which is normally 0. The boot command qualifier value of 6 indicates that bits 1 and 2 should be set.

  ```
  >>> boot -flag n,6 dev
  ```

This problem results in a kernel stack not valid error halt on a secondary CPU after you proceed from the second EXEC_INIT breakpoint using ;P.

To work around the problem, you must set also bit 17 in the boot flags, as shown in the following example:

```
>>> boot -flag n,20007 dev
```

In the example, the boot command qualifier value of 20007 indicates that bits 1, 2, and 17 will be set. Setting bit 17 avoids the kernel stack not valid error halt. It also causes a subset of the verbose boot messages to be displayed during the boot process.

This problem only occurs when a system boots from a SCSI disk and the boot command instructs the system to include XDELTA and to stop at the EXEC_INIT breakpoints. If the boot command includes XDELTA but does not specify stopping at the EXEC_INIT breakpoints, the problem does not occur.

The problem does not occur when a system boots from either a KDM70-based disk or from a CI based disk.

1.8 PNDRIVER and Setting of SCSSYSTEMID System Parameter — Problem

V1.5 If you are booting a new system disk and the value of the SCSSYSTEMID system parameter is 0, PNDRIVER receives an access violation. As a result, the system hangs.

To work around this problem, set the SCSSYSTEMID system parameter to a nonzero value.

Digital expects to fix this problem with PNDRIVER in a future release of OpenVMS AXP.

1.9 SCSI Disk Class Driver (DKDRIVER) — Restriction Removed

V2.0 The restriction on the SCSI disk class driver (DKDRIVER), that it did not support data check operations for OpenVMS AXP Version 1.0, has been removed. The comparison operations invoked by the IO$M_DATACHECK modifier, the IO$_WRITECHECK function code, and other requests for data check support now function correctly.

2

General User-Level Release Notes

This chapter contains information that is of interest to all users of the OpenVMS AXP operating system. It includes information about commonly used utilities, text processing utilities, and user-environment testing.

2.1 Release Notes Specific to OpenVMS AXP Version T6.1-FT4

V6.1 The release notes in this document are cumulative from OpenVMS AXP Version 1.0. In this chapter, the following sections contain general user release notes that pertain specifically to OpenVMS AXP Version T6.1-FT4:

- TBS

2.2 DCL Commands and Utilities

This section contains information about DCL commands and utilities.

2.2.1 DIFFERENCES Command Enhancements

V6.1 The DIFFERENCES utility has been enhanced to support the /IGNORE=CASE qualifier. This new qualifier causes the DIFFERENCES utility to do a case-blind comparison when comparing records from different files.

In addition, the DIFFERENCES utility has been enhanced to support the new /PAGE implementation described in Section 2.2.4.

2.2.2 DIRECTORY Command Enhancements

V6.1 The DIRECTORY command has been enhanced with the following qualifiers:

Qualifier	Action
/EXACT	Used with /SEARCH to do a case-sensitive search when string is enclosed in double quotes ("").
/HIGHLIGHT=[BOLD] [BLINK] [UNDERLINE] [REVERSE]	Used with /SEARCH to determine the style of highlighting. Options may be combined by separating them with commas.
/PAGE=SAVE[=*n*]	Use new PAGE implementation when outputting information. Save up to *n* screens of information. The default value for *n* is 5.
/SELECT=[NO]ACL]	Selects files for output that do [not] have ACLs associated with them.

Qualifier	Action
/SELECT=FILE=[[NO]NODE] [[NO]DEVICE] [[NO]DIRECTORY] [[NO]NAME] [[NO]TYPE] [[NO]VERSION]	Selects fields of the file specification to be output. All fields are output by default. Intended for use with the /BRIEF qualifier. Incompatable with the /FULL qualifier.

Also see Section 2.2.4 for a description of the new /PAGE=SAVE implementation.

2.2.2.1 Restrictions

V6.1 The /SELECT=FILE qualifier is incompatable with the /FULL qualifier.

2.2.3 EDIT Command

The following sections apply to the EDIT command.

2.2.3.1 EDT No Longer Default Editor

V6.1 As of Version Version T6.1-FT4 of OpenVMS AXP, the EDT editor and callable EDT are no longer system defaults. In the interest of upward compatibility, the EDT editor and callable EDT will continue to be provided, but without support, with OpenVMS AXP. Digital urges EDT users to migrate to EVE at their earliest convenience.

2.2.3.2 Change in Default Editor

V1.5 The default editor for the EDIT command has been changed to TPU from EDT. Therefore, when you enter the EDIT command, the Extensible Versatile Editor (EVE) is invoked rather than the EDT editor.

This change was made to provide better system performance. TPU is a native image on AXP systems, whereas the EDT editor is a translated image.

The EDIT command now invokes TPU with the EVE section file. Please see Section 2.3 for DECTPU and EVE release notes. See the *Guide to the DEC Text Processing Utility* and the *OpenVMS User's Manual* for more information about DECTPU and EVE.

Retaining the EDT Editor

To continue using the EDT editor, define the following symbol interactively or in your login command procedure:

```
$ EDIT :== EDIT/EDT
```

This symbol overrides the default and causes the EDIT command to invoke the EDT editor instead of EVE.

DCL Command Procedures and the EDIT Command

Verify that the /EDT qualifier is present in any DCL command procedure that relies on EDT. Any procedure that uses the EDIT command without the /EDT qualifier will fail because this verb now invokes TPU with the TPU$SECTION section file. (By default, this section file is EVE.)

EVE and the EDT Keypad

In EVE, you can select an EDT-like keypad by defining the AXP logical name EVE$KEYPAD to be EDT. See EVE documentation for details about how to do this at either the process level or system level.

System managers can find an example of how to define EVE$KEYPAD at the system level in the file SYS$STARTUP:SYLOGICALS.TEMPLATE.

2.2.4 /PAGE Qualifier—New Implementation

V6.1

A new implementation of the /PAGE qualifier has been provided for the following commands:

AUTHORIZE SHOW command
DIRECTORY
DIFFERENCES
DUMP
HELP
SEARCH
SHOW AUDIT
SHOW DEVICE
SHOW ENTRY
SHOW ERROR
SHOW LICENSE
SHOW LOGICAL
SHOW MEMORY
SHOW PROCESS
SHOW QUEUE
SHOW SYSTEM
SHOW USERS
TYPE

This implementation prompts you after one screen of information is displayed. In response you can press any one of the following keys:

Key	Action
RETURN	Continue outputting lines.
ENTER	Same as RETURN.
SPACE	Same as RETURN unless in HELP mode.
CTRL/Z	Exit utility.
F10	Same as CTRL/Z.
LEFT-ARROW	Display last column scrolled off right of screen.
RIGHT-ARROW	Display last column scrolled off left of screen.
DOWN-ARROW	Display last line scrolled off bottom of screen.
UP-ARROW	Display last line scrolled off top of screen.
CTRL/W	Repaint the screen.
FIND (E1)	Change search string. (Not supported with the SEARCH command.)
INSERT-HERE (E2)	Display last group of columns scrolled off left of screen.
REMOVE (E3)	Display last group of columns scrolled off right of screen.
SELECT (E4)	Toggle 80/132 column mode.
PREV-SCREEN (E5)	Display last page scrolled off top of screen.
NEXT-SCREEN (E6)	Display last page scrolled off bottom of screen.

Key	Action
HELP (F15)	Display help on specified topic.
DO (F16)	Toggle display to oldest/newest page.
Any others	Ignored (except for the TYPE command - see *OpenVMS DCL Dictionary: N–Z* for details).

If there is no more text to be displayed in the direction requested, the terminal bell sounds.

When PAGE mode is used, the following qualifiers are also allowed:

Qualifier	Action
/EXACT	Used with /SEARCH to do a case-sensitive search when string is enclosed in double quotes ("").
/HIGHLIGHT=[BOLD] [BLINK] [UNDERLINE] [REVERSE]	Used with /SEARCH to determine the style of highlighting. Options may be combined by separating them with commas.
/PAGE[=SAVE[=*n*]]	Use PAGE implementation when outputting information. Save up to *n* screens of information. The default value for *n* is 5.
/SEARCH=*string*	Highlight any line containing the string *string*. (Not supported with the SEARCH command.)

2.2.4.1 Restrictions

V6.1

The following restrictions apply to the /PAGE qualifier:

- The SEARCH command does not support the /SEARCH qualifier or the FIND key.
- The TYPE command requires the /PAGE=SAVE qualifier to use the new page implementation. Specifing TYPE/PAGE will result in the old page behavior, which is compatable with previous versions of OpenVMS.
- The TYPE/PAGE=SAVE command is implemented using the OpenVMS Screen Management Run-Time Library (SMG$). Since SMG$ does not support outputting of all possible escape sequences, files containing escape sequences that are output with the TYPE/PAGE=SAVE command may not appear to be output correctly. This is a permanent SMG$ restriction.

2.2.5 PRINT/NOTIFY and SUBMIT/NOTIFY Commands

V6.1

In OpenVMS Version 5.5-2, a problem existed with the PRINT/NOTIFY and SUBMIT/NOTIFY commands. When you printed or submited a job to a generic queue with the /NOTIFY qualifier, the notification message incorrectly displayed the name of the *generic* queue on which the job completed. This behavior was inconsistent with versions earlier than OpenVMS Version 5.5, where the notification message displayed the name of the *execution* queue on which the job completed.

This problem has been corrected. When you print or submit a job to a generic queue with the PRINT/NOTIFY or SUBMIT/NOTIFY command, the notification message displays the name of the execution queue on which the job completes.

2.2.6 SET PROCESS/SUSPEND=KERNEL/ID= Command in a Cluster Environment — Incorrect Behavior

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 4.32.8. Digital expects to fix this problem in a future version of OpenVMS AXP.

2.2.7 SET TERMINAL Command Enhancements

V6.1

The SET TERMINAL command has been enhanced as follows:

- When the /PAGE=n qualifier is specified to set the terminal page length and the DEC_CRT4 characteristic is set, the SET TERMINAL command now sends the appropriate escape sequence to change the number of lines per screen.
- When the /INQUIRE or /DEVICE=n qualifier is specified and the DEC_CRT characteristic is set, the SET TERMINAL command now reads the current screen size from the terminal and sets the corresponding page length and page width values appropriately. With this enhancement DECWINDOWS DECterm windows are no longer resized by SET TERMINAL/INQUIRE.

2.2.7.1 Restrictions

V6.1

The enhancements to the SET TERMINAL/INQUIRE command work correctly on DIGITAL supplied VT100 and later terminals. Some Personal Computer terminal emulators may not work correctly since they do not correctly emulate all VT100 escape sequences. If you experience problems with these terminal emulators, DIGITAL recomends that you contact your terminal emulator supplier.

2.2.8 TYPE Command Enhancements

V6.1

The TYPE command has been enhanced to be compatable with OpenVMS VAX Version 6.0. These enhancements include support for the following qualifiers:

Qualifier	Action
/CONTINUOUS/INTERVAL=*n*	Upon reaching the end of a file, wait *n* seconds (the default is 30) then display any records that have been added to the file.
/PAGE=CLEAR_SCREEN	Clear screen before each new page of text.
/PAGE=SCROLL	Scroll the screen one line at a time when outputting each new page of text.
/TAIL[=*n*]	Output the last *n* lines of the file.

The following qualifiers are OpenVMS AXP additions:

Qualifier	Action
/EXACT	Used with /SEARCH to do a case-sensitive search when string is enclosed in double quotes ("").

Qualifier	Action
/[NO]HEADER	Controls output of the header line that is printed before each file.
/HIGHLIGHT=[BOLD] [BLINK] [UNDERLINE] [REVERSE]	Used with /SEARCH to determine the style of highlighting. Options may be combined by separating them with commas.
/PAGE=SAVE[=*n*]	Use new PAGE implementation when outputting information. Save up to *n* screens of information. The default value for *n* is 5.
/SEARCH=*string*	Highlight any line containing the variable *string*.
/WRAP	Used with /PAGE=SAVE to force lines wider than the screen width to be wrapped to the next line. By default lines are not wrapped and the INSERT key must be used to scroll the screen to the left.

See Section 2.2.4 for a description of the new /PAGE=SAVE implementation.

In addition the logical TYPE$DEFAULT_FILESPEC is used to determine the default file specification used by the TYPE command. If not defined, the default value used is ".LIS". This is compatable with previous versions of OpenVMS.

2.2.8.1 Restrictions

V6.1

The following restrictions apply to the /TYPE qualifier:

- The TYPE command requires /PAGE=SAVE to use the new page implementation. Specifing TYPE/PAGE will result in the old page behavior that is compatable with previous versions of OpenVMS.
- TYPE/PAGE=SAVE is implemented using the OpenVMS Screen Management Run-Time Library (SMG$). Since SMG$ does not support outputting of all possible escape sequences, files containing escape sequences that are output with the TYPE/PAGE=SAVE command may not appear to be output correctly. This is a permanent SMG$ restriction.
- TYPE/TAIL has the following restrictions:
  - No null characters in records
  - Sequential file organisation only
  - STREAM, STREAM_LF, STREAM_CR, VFC and VAR record formats only
  - Maximum record size 511 bytes

2.3 DECTPU and EVE Version 3.1 Notes

On AXP systems, the default editor is EVE. See Section 2.2.3.2 for information about this change in default editors.

2.3.1 DECwindows Motif for OpenVMS AXP Operations from EVE in a DECterm Emulator

V1.0

When EVE is used in a DECterm emulator, the mouse buttons are handled and processed by the underlying TPU code. This allows you to define the mouse buttons and bind TPU code to them. If you want to perform normal DECwindows Motif for OpenVMS AXP cut and paste operations from the DECterm emulator in which EVE is executing, you must turn off TPU's mouse support. The following command, when used from within EVE, allows normal DECwindows Motif for OpenVMS AXP cut and paste operations to occur:

```
Command: tpu set (mouse,off)
```

This command can be entered interactively or added to an EVE initialization file.

If you want to switch between TPU mouse support and DECwindows Motif for OpenVMS AXP cut and paste operations, add the following two procedures to a TPU command file:

```
procedure eve_mouse_off;
set (mouse,off);
endprocedure;
procedure eve_mouse_on;
set (mouse,on);
endprocedure;
```

After the two commands have been compiled, you can then enter the command `Command: Mouse off` or `Command: Mouse on` to toggle between TPU and DECwindows Motif for OpenVMS AXP mouse support.

See the *OpenVMS User's Manual* for more information on how to add new EVE commands and define the mouse buttons.

2.3.2 Motif Widget Context Help Built-In

V1.0 The following built-in enters the Motif context-sensitive help mode:

```
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 or does not confine the pointer if OFF or 0.

Note

This built-in is disabled due to a problem in the Motif toolkit.

2.3.3 Translated Images Using the TPU Callable Interface

V6.1 Translated images that used the TPU callable interface could create an access violation if they specified callback routines when initializing TPU.

This problem has been fixed.

2.4 Sort/Merge Utility

V1.5 The Sort/Merge utility (SORT/MERGE) runs as a native AXP image. For Version 1.5, common data dictionary support is included. CDD/Repository is available on OpenVMS AXP. All user-visible functions and interfaces of SORT/MERGE are identical to OpenVMS VAX SORT/MERGE, with the following exceptions:

- Support for IEEE S and T floating-point datatypes is now available. Two new qualifiers, S_FLOATING and T_FLOATING, have been added for the /KEY qualifier to specify these datatypes, as shown in the following example:

```
SORT /KEY=(POS:1,S_FLOATING) INFILE.DAT OUTFILE.DAT
```

The qualifier S_FLOATING specifies S_FLOATING format data in the key field. The qualifier T_FLOATING specifies T_FLOATING format data in the key field. The SIZE parameter is optional. However, if SIZE is specified, it must be 4 for S_FLOATING or 8 for T_FLOATING data.

- SORT does not trap IEEE NaNs or Denormals. Instead, they are collated as follows:

-NaN	< -Infinity
+NaN	> +Infinity
-Denormal	< 0 but greater than any other negative number
+Denormal	> 0 but less than any other positive number

- During the course of SORT/MERGE development, the specification file format changed from the VAX VMS Version 3 format to the current format. If you still have specification files that need to be converted from VAX VMS Version 3.0 format to the current format, you must use the SRTTRN.EXE utility on VAX VMS Version 4.0 through Version 6.0. SRTTRN is not provided on OpenVMS AXP. This is a permanent restriction.
- The behavior for reporting an invalid digit in a decimal string input differs between VAX systems and AXP systems.

  On VAX systems, you receive a reserved operand fault and a message that the value has been converted to a valid decimal string digit.

  On AXP systems, there is no indication of an invalid decimal digit or that a conversion has occurred. AXP systems do not have a decimal string hardware instruction to generate such a fault. The decimal string arithmetic is emulated in software. The data is converted with a decimal string zero inserted for the invalid character for comparison purposes. However, the data in the output file remains identical to that in the input file.
- When SORT creates an output file containing fixed-length records whose record lengths vary from those of the input records, the following warning message is issued the first time a record with a different length is written:

```
%SORT-W-VAR_FIX, records may be truncated or padded in output file:
        device:[user_directory]filename.ext
```

  Previously, SORT issued an "invalid record size" error message for every input record that was truncated or padded on output.

3 System Management Release Notes

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

3.1 Release Notes Specific to OpenVMS AXP Version T6.1-FT4

V6.1 The release notes in this document are cumulative from OpenVMS AXP Version 1.0. In this chapter, the following sections contain system management release notes that pertain specifically to OpenVMS AXP Version T6.1-FT4:

- TBS

For additional information about system management features and tasks, refer to *A Comparison of System Management on OpenVMS AXP and OpenVMS VAX.* The Comparison is intended for experienced system managers who need to learn quickly how specific management tasks differ or remain the same on AXP and VAX computers.

3.2 Modifications to Analyze/Disk_Structure Utility

V1.5 The following two qualifiers have been added to the Analyze/Disk_Structure utility:

- /ATTRIBUTES

  The /ATTRIBUTES qualifier determines if the Analyze/Disk_Structure utility repairs files containing erroneous settings in the record attributes section of their associated File Attribute Blocks (FAB).

  The /ATTRIBUTES qualifier may be used with the /REPAIR qualifier. If attribute repair is enabled during the repair phase, erroneous bits are cleared from a file's record attributes. This action may not correctly set a file's record attributes because it is beyond the scope of this utility to determine their correct values.

  Digital recommends that system managers do not perform an attribute repair; instead they should notify the owners of the files about the inconsistencies and have the owners reset the files' attributes using the SET FILE/ATTRIBUTES=({*record-attributes*}) command.

- /OUTPUT

  The /OUTPUT qualifier specifies the output file to which the disk structure errors are written. If you omit the /OUTPUT qualifier or the file specification, the output is directed to SYS$OUTPUT. If /NOOUTPUT is specified, no disk structure errors are displayed. If the /CONFIRM qualifier is specified, output is forced to SYS$OUTPUT regardless of the use of /OUTPUT or /NOOUTPUT.

For more information about the Analyze/Disk_Structure Utility, see the *OpenVMS System Management Utilities Reference Manual.*

3.3 Adding Special Accounts

V6.1 Effective with this release, the username SYSTEM is the only one that is automatically created as part of a fresh installation. In order to run the User Environment Test Program (UETP) usernames SYSTEST and SYSTEST_CLIG must be created. Some sites may also want to create a Field Service account for use by regularly assigned Field Service personnel.

To create accounts for UETP or Field Service, a new command procedure is provided which can be invoked with the command:

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

In contrast to former practice, this command procedure allows choice of the username for Field Service personnel. This allows assignment of separate usernames to separate individuals for better accountability and also reduces the risk of successful outside attacks against a known username such as FIELD.

Digital still recommends that the privileged accounts described above be disabled when not in use and that generated passwords be used for Field Service accounts.

3.4 Authorize Utility

V1.0 Authorize utility commands and parameters are identical on AXP and VAX systems. However, the default values for a number of AXP process limits and quotas are higher. See *A Comparison of System Management on OpenVMS AXP and OpenVMS VAX* for more information.

Digital reserves certain group user identification code (UIC) values on both AXP and VAX operating systems. When specifying a UIC with the /UIC qualifier, valid UIC values for the group number are 2 to 277 (octal) and 400 to 37776 (octal). Group 1 and groups 300 to 377 are reserved.

3.5 AUTOGEN Command Procedure

V1.0 AUTOGEN automatically sizes system parameters after you install or upgrade the operating system and after you install optional layered products. Digital recommends that you run AUTOGEN on a weekly basis in order to adjust your system parameters according to your system's workload.

See the *OpenVMS AXP Version 6.1 Upgrade and Installation Manual* for information about running AUTOGEN after installing the operating system. See *A Comparison of System Management on OpenVMS AXP and OpenVMS VAX* and the *OpenVMS System Manager's Manual* for additional information about using AUTOGEN.

3.5.1 Problem During System Startup

V1.0 There have been instances where operational problems sometimes cause the operating system to enter a loop similar to the following:

1. During system startup, the startup procedure reports that DECwindows cannot be loaded because some system parameters necessary for DECwindows operation are set incorrectly. The system startup procedure asks if AUTOGEN should be run in order to correct the problem.
2. After you answer yes, the AUTOGEN procedure runs.
3. The system then automatically reboots.

4. The system again reports that the DECwindows-related system parameters are set incorrectly.

If your system enters a loop as described above, consider the following possible problems:

- In some instances, system parameter settings related to DECwindows have not taken effect because the automatic reboot used a different root than the previous boot.
- In some cases, the values in MODPARAMS.DAT force AUTOGEN to set system parameters related to DECwindows incorrectly.

If the system startup procedure reports that system parameter settings related to DECwindows are incorrect after an automatic reboot, confirm that the boot is being done from the proper root. In addition, verify that the parameter settings in MODPARAMS.DAT do not conflict with DECwindows requirements.

3.5.2 Support for Installed Resident Images

V1.5 AUTOGEN has been modified to update the GH_RSRVPGCNT and ITB_ENTRIES system parameters if feedback is specified or if these parameters are explicitly specified in MODPARAMS.DAT.

When feedback is requested, the GH_RSRVPGCNT parameter is modified based on how many pages were allocated from the granularity hint code region after system startup has finished. AUTOGEN sets GH_RSRVPGCNT to the smaller value of either its current value or the number of pages (pages allocated after system startup) plus an additional 10% in order to leave slightly more room for expansion. After AUTOGEN calculates a new value for GH_RSRVPGCNT, it then modifies ITB_ENTRIES accordingly.

The GH_RSRVPGCNT and ITB_ENTRIES parameters are updated by AUTOGEN unless any of the following conditions exist:

- Feedback is not specified and the GH_RSRVPGCNT and ITB_ENTRIES parameters are not explicitly set in MODPARAMS.DAT.
- Feedback is specified, but the SGN$V_RELEASE_PFNS flag is not set in the LOAD_SYS_IMAGES system parameter, which indicates that no PFNs are to be released from the granularity hint regions. Therefore, there is no need to modify these system parameters.
- Feedback is specified, but the number of pages currently allocated in the granularity hint code region is less than the number of pages allocated at the end of system startup.

If any of those conditions exist, AUTOGEN does not modify the GH_RSRVPGCNT and ITB_ENTRIES system parameters.

3.5.3 Booting a node into a VMScluster—Problem

V6.1 The following error messages may be seen when booting a node into a VMScluster:

```
waiting to form or join a VMScluster system
%VMScluster-I-LOADSECDB, loading the cluster security database
%CNXMAN,  Proposing formation of a VMScluster
%CNXMAN,  Now a VMScluster member -- system VMSTS5
%CNXMAN,  Completing VMScluster state transition
%MSCPLOAD-I-CONFIGSCAN, enabled automatic disk serving
$!  Copyright (c) 1993 Digital Equipment Corporation.  All rights reserved.
%STDRV-I-STARTUP, VMS startup begun at  5-AUG-1993 14:04:00.11
%INSTALL-E-RESFAIL, failed to install image with /RESIDENT qualifier
-INSTALL-E-NOGHREG, insufficient memory in the code granularity hint region
```

These messages indicate that not all images could be installed resident. This can result in some level of performance degradation, but is otherwise benign.

A possible workaround is to increase the values of the ITB_ENTRIES and GH_RSRVPGCNT system parameters.

3.5.4 System Parameters

V1.0

Digital recommends that you use AUTOGEN to modify system parameters. You can also modify system parameters by using the System Management utility (SYSMAN) or by using the System Generation utility (SYSGEN).

3.5.4.1 AWSTIME and QUANTUM System Parameters

V1.0

Some application configurations (for example, a large number of memory-intensive processes) may benefit if the values of the AWSTIME and QUANTUM system parameters are reduced. The value for both parameters should be identical and can be as low as 4.

3.5.4.2 DUMPSTYLE System Parameter — New Values

V1.5

The DUMPSTYLE system parameter lets you control the amount and type of information that is generated when an OpenVMS AXP system fails.

When an OpenVMS AXP system fails, the system first sends information to the operator's console terminal (if one exists) and then writes the memory contents either fully or selectively to the system dump file. (If there is no console terminal, you do not see the console data.)

Both sets of information can be quite complex and very long. However, with the DUMPSTYLE system parameter, you can control how much information you receive, as shown in Table 3–1.

Table 3–1 DUMPSTYLE System Parameter Values

Value	Meaning
0	Full dump with minimal console output
1	Selective dump with minimal console output
2	Full dump with full console output
3	Selective dump with full console output

The DUMPSTYLE system parameter for Version 1.5 introduces options for the amount of console output. Digital retained the values 0 and 1 for the shorter console output on AXP systems so that you do not have to change your DUMPSTYLE system parameter to get the default, shorter output. The new values of 2 and 3 let users select both the amount of information that is written to the system dump file and the amount of information that is displayed as console output. The default value is 1.

On an AXP system, the SHOW DUMPSTYLE command in the SYSGEN utility lists the default value for the DUMPSTYLE system parameter as 0. However, the AUTOGEN calculated value (effectively a default) is 1.

3.5.4.3 MULTIPROCESSING System Parameter

V1.5 The MULTIPROCESSING system parameter, which controls loading of the system synchronization image, has a new value of 4. The new value guarantees that the streamlined image will be loaded in situations where the uniprocessor image would have been loaded by default.

The Alpha AXP architecture allows the possibility of adding CPUs to the configuration set after a system boot. The new system parameter value of 4 lets system managers boot a single CPU symmetric multiprocessing (SMP) configuration and then add additional, previously unavailable CPUs (not allowed under VAX systems). The default value for the MULTIPROCESSING parameter will continue to be the same as for VAX systems; the new parameter value has been added to allow the new capability. The system parameter's values have equivalent functions on AXP and VAX systems. Table 3–2 summarizes the functions of the five MULTIPROCESSING values.

Table 3–2 MULTIPROCESSING Values on AXP and VAX Systems

Value	Function
0	Load uniprocessing synchronization image.
1	Load full-checking multiprocessing synchronization image if CPU type is capable of SMP and two or more CPUs are present on the system.
2	Always load full-checking version, regardless of system configuration or CPU availability.
3	Load streamlined multiprocessing synchronization image if CPU type is capable of SMP and two or more CPUs are present on the system.
4	Always load streamlined multiprocessing synchronization image, regardless of system configuration or CPU availability.

The new parameter value of 4 is comparable to the value of 2 for the full-checking system synchronization image. When the value is set to 4, even uniprocessor configurations load the multiprocessing system synchronization image, but in this case the streamlined one, rather than the full-checking one.

When the full-checking multiprocessing synchronization image is loaded, OpenVMS performs software sanity checks on the node's CPUs; in addition, OpenVMS provides a history of spinlock acquisition information for each spin lock. This historical information is available in the event of a system failure. OpenVMS stores a program counter (PC) history in the spinlock (SPL) structures used to synchronize system activity. When the system fails, that information is accessible by using the System Dump Analyzer utility (SDA) SHOW SPINLOCK command. The information displayed includes the PCs of the last 16 acquisitions and releases of the spin locks.

An SMP node running the full-checking image performs more slowly than a node running the streamlined image. It is easier to debug system failures on SMP nodes, though, when the full-checking image is enabled. The streamlined image is designed for faster performance, with the tradeoff being less extensive debug support following a system failure.

See Section 3.36 and *A Comparison of System Management on OpenVMS AXP and OpenVMS VAX* for more information about symmetric multiprocessing support in Version 1.5.

3.5.4.4 PHYSICAL_MEMORY System Parameter

V1.0 On AXP systems, the PHYSICAL_MEMORY system parameter replaces the PHYSICALPAGES system parameter found on VAX systems. PHYSICAL_MEMORY is measured in megabytes.

To reduce the amount of physical memory available for use, change the PHYSICAL_MEMORY parameter. The default value for the PHYSICAL_MEMORY parameter is the maximum value allowed. See the *OpenVMS System Manager's Manual* for information about how to calculate the proper system dump file size for your system.

3.5.4.5 System Parameters and Performance Measurements

V1.0 The chapter on performance optimization tasks and Appendix A in *A Comparison of System Management on OpenVMS AXP and OpenVMS VAX* contain information about system-tuning tips that can be used to measure and enhance the performance of your AXP system.

3.5.4.6 System Parameters That Have Been Removed

V1.0 The following system parameters are not present on AXP systems:

IRPCOUNT
IRPCOUNTV
LRPCOUNT
LRPCOUNTV
LRPSIZE
SRPCOUNT
SRPCOUNTV

These parameters were removed as part of the changes for adaptive pool management. Adaptive pool management offers simplified system management, improved performance, and reduced overall pool memory requirements. See *A Comparison of System Management on OpenVMS AXP and OpenVMS VAX* for more information about adaptive pool management.

3.5.5 AUTOGEN Tuning Problem

V1.5 AUTOGEN may incorrectly set the system parameter PFCDEFAULT to 128 on satellite nodes in a cluster. This can increase the number and size of paging I/Os to the boot node.

To prevent this problem, insert the following line in the SYS$SYSTEM:MODPARAMS.DAT file before you run AUTOGEN:

```
PFCDEFAULT=64
```

3.5.6 SETPARAMS Phase — Behavior Change

V1.5 The AUTOGEN SETPARAMS phase behaves differently on an OpenVMS AXP system than it does on an OpenVMS VAX system.

On OpenVMS VAX, the AUTOGEN SETPARAMS phase resets all system parameters to their default values. AUTOGEN then modifies some or all of those values by using data gathered in feedback mode or by using user-supplied data contained in the MODPARAMS.DAT file. If no modification is made, the default value is used the next time the system boots.

On OpenVMS AXP, the AUTOGEN SETPARAMS phase uses the values from the current system image; values are not reset to their defaults. Current values are used, except for those that are explicitly modified by AUTOGEN.

For more information about how AUTOGEN uses default values, see the AUTOGEN Help topic EXECUTION-TYPE, INITIAL subtopic.

3.6 Backup Utility

V1.5

The Backup utility on AXP systems now supports hardware data compaction for SCSI tape drives that provide this capability. Use the BACKUP command qualifier /MEDIA_FORMAT=COMPACTION for these devices.

3.6.1 Change in Processing ALIAS Directory Trees

V6.1

In previous versions of VAX and AXP OpenVMS BACKUP when a /IMAGE or /FAST save of a disk was done, ALIAS directory trees were considered for processing even though the primary file had already been saved. Also, the first attempt to process a file occurred even if the ALIAS directory tree was being scanned. To prevent this, users used the /EXCLUDE qualifier to prevent processing of ALIAS directory trees, for example:

```
/EXCLUDE=([sys*.syscommon...]*.*;*)
```

In OpenVMS VAX Version T6.1-FT4 and OpenVMS AXP Version T6.1-FT4, when a /IMAGE or /FAST save is done of a disk, ALIAS directory trees are not processed. Only the primary files that the ALIAS points to are saved. Depending on the number of ALIAS directory specifications there are on the disk, this may increase performance by reducing the number of files BACKUP checks for processing. If the /FAST qualifier is used for the save operation, a message will be displayed for each ALIAS directory or file that is not processed.

3.6.2 BACKUP Now Extends Index Files

V6.1

Problem:

In previous versions of the operating system, a problem occurred when you initialized a disk and extended the INDEXF.SYS file using the /HEADERS and /MAXIMUM_FILES qualifiers. During a BACKUP/IMAGE restore or copy operation with /NOINITIALIZE as the output-device qualifier, BACKUP ignored the extension that you made to the index file. BACKUP created on the target disk an INDEXF.SYS file that was the same size as the INDEXF.SYS file on the source disk.

With OpenVMS AXP Version 1.5, BACKUP preserved the size of the INDEXF.SYS file on the target disk when you used /NOINITIALIZE as an output-device qualifier. As a result, the index file on the target disk had to be large enough to accommodate the number of files copied from the source disk. If INDEXF.SYS was too small, the backup operation aborted and displays an error message.

With OpenVMS AXP Version T6.1-FT4, BACKUP preserves the INDEXF.SYS file of the target disk if it is large enough to accommodate the number of files copied from the source disk. If the INDEXF.SYS file is too small, BACKUP extends the INDEXF.SYS file to the optimal size and displays a message.

3.6.3 Change to BACKUP/VERIFY

V6.1

With previous versions of the OpenVMS operating system, BACKUP did not perform verification on the output files in a disk-to-disk copy operation. The BACKUP/VERIFY command displayed a message indicating that it was performing verification on the output files, but actually it was not.

OpenVMS AXP Version T6.1-FT4 corrects this problem by adding a second verify file scanning pass to compare the input and output files.

Note

BACKUP/LOG displays messages as files are successfully saved or verified. Because BACKUP displays the messages in the order that these actions occur, the sequence of the messages may be different from that of previous versions of OpenVMS BACKUP.

3.6.4 BACKUP/EXACT ACCVIO Problem Fixed

V6.1

In previous versions of the OpenVMS operating system, a BACKUP command with all of the following characteristics caused an access violation (ACCVIO):

- The command specified the /EXACT_ORDER qualifier.
- The command did not specify magnetic tape labels (using the /LABEL qualifier).
- The command used more than 20 magnetic tape volumes in a single BACKUP operation.

OpenVMS AXP Version T6.1-FT4 corrects this problem.

3.6.5 BACKUP/LIST Enhancement

V6.1

In previous versions of the OpenVMS operating system, the BACKUP/LIST command displayed pre-allocated files as using zero blocks of disk space. Pre-allocated files have a zero end-of-file block (EFBLK), but their allocated size is greater than zero. The DIRECTORY/SIZE=ALL and DIRECTORY/FULL commands correctly display the same number of blocks used as allocated.

OpenVMS AXP Version T6.1-FT4 corrects this problem. The BACKUP/LIST command now displays the same number of blocks used as allocated for pre-allocated files.

3.6.6 BACKUP/STORAGE_MANAGEMENT Qualifier Change

V6.1

With previous versions of OpenVMS operating system, BACKUP accepted the /STORAGE_MANAGEMENT qualifier regardless of whether the Storage Library System (SLS) product was installed. This could eventually result in various BACKUP errors.

OpenVMS AXP Version T6.1-FT4 corrects this problem. BACKUP now rejects the /STORAGE_MANAGEMENT qualifier if SLS is not installed.

3.6.7 Changes to BACKUP Input File Processing

The release notes in this section describe fixes to BACKUP input file processing.

3.6.7.1 Relative File Version Support

V6.1 With previous versions of the OpenVMS operating system, BACKUP did not save input files that were specified by using relative file versions. For example:

```
$ BACKUP A.A;-1,A.A;0,A.A;-5 SAVE.BCK/SAVE
```

OpenVMS AXP Version T6.1-FT4 corrects this problem.

Note

BACKUP processes the relative version -0 as 0, selecting the most recent version of the file for processing. Digital expects to correct this problem in a future release.

3.6.7.2 Rooted Directory Specification Support

V6.1 With previous versions of the OpenVMS operating system, BACKUP did not save input files that were specified using a rooted directory specification. For example:

```
$ BACKUP DEV:[DIR.][SUBDIR]*.*;* SAVE.BCK/SAVE
```

OpenVMS AXP Version T6.1-FT4 corrects this problem.

3.6.7.3 BACKUP Handling of Forced Errors Improved

V6.1 With previous versions of the OpenVMS operating system, files which received Forced Error read errors during an OpenVMS BACKUP image save operation would be included in a saveset twice, once in the original directory and once as part of the lost files pass. This wasted save set space and could cause a restore operation to fail due to restoring more files and disk blocks than was originally contained on the source disk.

Additionally, the restore operation did not indicate that an error occurred during the save operation and that the restored file's data may be questionable.

OpenVMS AXP Version T6.1-FT4 corrects this problem. BACKUP issues the appropriate error messages but does not include files a second time in the saveset as lost files. Additionally, BACKUP now issues a warning during the restore operation regarding any errors that occurred during the save operation.

3.6.8 Audit Alarms Reduced

V6.1 With previous versions of the OpenVMS operating system, the way in which BACKUP accessed the master file directory (MFD) sometimes generated unnecessary audit alarms.

OpenVMS AXP Version T6.1-FT4 greatly reduces the number of these audit alarms by accessing the MFD differently.

3.6.9 NOMSG Problem Corrected

V6.1 With previous versions of OpenVMS operating system, BACKUP sometimes displayed the NOMSG error message while trying to display a message.

OpenVMS AXP Version T6.1-FT4 corrects this problem.

3.6.10 Tape Handling Problem Corrected

V6.1 With previous versions of the OpenVMS operating system, when a BACKUP save operation to tape failed with a PROCINDEX error message, you could not append any save sets to the tape (BACKUP issued the NOTANSI error message).

OpenVMS AXP Version T6.1-FT4 corrects this problem.

3.6.11 Restrictions

V1.5 The following restrictions apply to the Backup utility:

- A BACKUP operation to mixed tape and disk save sets, as shown in the following command, is unsupported:

```
$ BACKUP SYS$DISK:/IMAGE dka0:FOO,MKA0:/SAVE/REW
```

V6.1 On an OpenVMS system disk, the file [SYSx]SYSCOMMON.DIR is an alias directory of the file [000000]VMS$COMMON.DIR. This means that both files point to the same file header. Prior to OpenVMS AXP Version 1.5 operating system, during the restore operation, BACKUP did not properly restore the VMS$COMMON.DIR file. Although this does not affect the system disk, it might produce errors with Digital Command Language (DCL) lexical functions.

OpenVMS AXP Version 1.5 corrected this problem. However, if you restored image backups created with a previous OpenVMS version, the problem recurred.

The symptoms of the problem are different depending on which version of the operating system you are using. If you upgraded from OpenVMS AXP Version 1.5 to OpenVMS AXP Version T6.1-FT4, it is unlikely that your system disk has this problem. However, you should confirm this and correct the problem if necessary.

Upgrading from OpenVMS AXP Version 1.0 Systems

To determine if your system disk has this problem, make sure that you have the LOGIO privilege and enter the DUMP/HEADER/BLOCK command as follows:

```
$ DUMP/HEADER/BLOCK=(COUNT=0) DISK:[000000]VMS$COMMON.DIR

  Map area offset:                    100
   Access control area offset:        255
   Reserved area offset:              255
  .
  .
  .
  Identification area
      File name:                    SYSCOMMON.DIR;1
      Revision number:              3
      Creation date:                15-JUN-1989 05:27:35.68
      Revision date:                23-JUN-1992 13:13:53.04
      Expiration date:              <none specified>
      Backup date:                  <none specified>

  Map area

      Retrieval pointers
          Count:          2       LBN:       5411

  Checksum:                              16366
```

If the file name in the Identification Area part of the display is not VMS$COMMON.DIR, as shown in this example, your system disk is affected by this problem. To correct this problem follow the procedure in Fixing the System Disk.

Upgrading from OpenVMS AXP Version 1.5 Systems

If you upgraded to OpenVMS AXP Version T6.1-FT4 from OpenVMS AXP Version 1.5 without following the procedure in Fixing the System Disk, 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-1992 19:31
[]000000.DIR;1                                           1  24-SEP-1992 19:31
[]SYSCOMMON.DIR;1                                        2  24-SEP-1992 19:31
[]SYSLIB.DIR;1                                          18  24-SEP-1992 19:31
[]SYSTEST.DIR;1                                          1  24-SEP-1992 19:31
[]SYSMAINT.DIR;1                                         1  24-SEP-1992 19:31
[]SYSMGR.DIR;1                                           6  24-SEP-1992 19:31
[]SYSHLP.DIR;1                                           6  24-SEP-1992 19:31
[]EXAMPLES.DIR;1                                         1  24-SEP-1992 19:31
[]SYSUPD.DIR;1                                           4  24-SEP-1992 19:31
[]SYSMSG.DIR;1                                           3  24-SEP-1992 19:31
     .
     .
     .
[]SECURITY_AUDIT.AUDIT                                   2   3-AUG-1993 15:23
[]SECURITY_AUDIT.AUDIT                                  11   3-AUG-1993 15:23
[]BACKUP.EXE;33                                        273   4-AUG-1993 09:37
[]STABACKUP.EXE;9                                      486   4-AUG-1993 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 the previous example), your system disk is affected by this problem. To correct this problem follow the procedure in Fixing the System Disk.

Fixing the System Disk

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
```

Note

Restoring image backups created with a previous VMS version will cause the problem to recur.

V6.1

- By using the /COMPARE and /IMAGE qualifiers, you can instruct BACKUP to perform an image compare operation. The image compare operation compares the files on one disk to the files on a second disk using the file identifications FIDs of the files.

  An image compare operation sometimes does not work correctly when you create two disks with identical files by incrementally backing up and restoring the files from one disk to the other disk. This is because BACKUP does not insure that the incrementally restored files have the same FIDs as the incrementally saved files. This is true regardless of whether the /OVERLAY, /NEW_VERSION, or /REPLACE qualifiers were used in the restore command.

3.7 Standalone BACKUP Not Supported

V6.1

Standalone BACKUP has been replaced by a menu-driven procedure included with the OpenVMS AXP Version T6.1-FT4 operating system. You start this procedure by booting the OpenVMS AXP Version T6.1-FT4 operating system distribution compact disc (from your local drive or from the InfoServer). After booting, you can choose from menu options to perform the following tasks:

- Enter a DCL environment, from which you can perform backup and restore operations on the system disk (instead of using standalone BACKUP)
- Install or upgrade the operating system, using the POLYCENTER Software Installation utility

For more detailed information about using the new menu-driven procedure included on the OpenVMS AXP operating system distribution compact disc, see the *OpenVMS AXP Version 6.1 Upgrade and Installation Manual.*

3.8 Batch and Print Queueing System

The following notes apply to the new batch and print queueing system introduced in OpenVMS AXP Version 1.5.

3.8.1 PRINT/DELETE Command Requirement

V1.5

Before OpenVMS AXP 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 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 OpenVMS AXP 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.*

The information in this note also applies to the SUBMIT/DELETE command.

3.8.2 STOP/QUEUE/RESET Command —Problem

V1.5 Under certain conditions, executing the START/QUEUE command within 1 minute of executing the STOP/QUEUE/RESET command can cause a symbiont process and all queues serviced by the symbiont to stop. When this happens, the following messages are written to the operator log:

```
%%%%%%%%%%%  OPCOM  20-JAN-1993 09:53:29.20  %%%%%%%%%%%
Message from user QUEUE_MANAGE on VMSPRT
%QMAN-E-SYMDEL, unexpected symbiont process termination

%%%%%%%%%%%  OPCOM  20-JAN-1993 09:53:29.21  %%%%%%%%%%%
Message from user QUEUE_MANAGE on VMSPRT
-PSM-F-BADLOGIC, internal logic error detected at PC 00000000
```

To avoid this problem, if you enter a STOP/QUEUE/RESET command, wait at least 1 minute before entering the START/QUEUE command.

3.9 Using DECdtm Services in a DECnet/OSI Network

V6.1 Read this section if all the following apply to you:

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

SCSNODE is a system parameter that defines the name of a computer. DECdtm transactions may fail if the SCSNODE value for a computer is the same as other computer names.

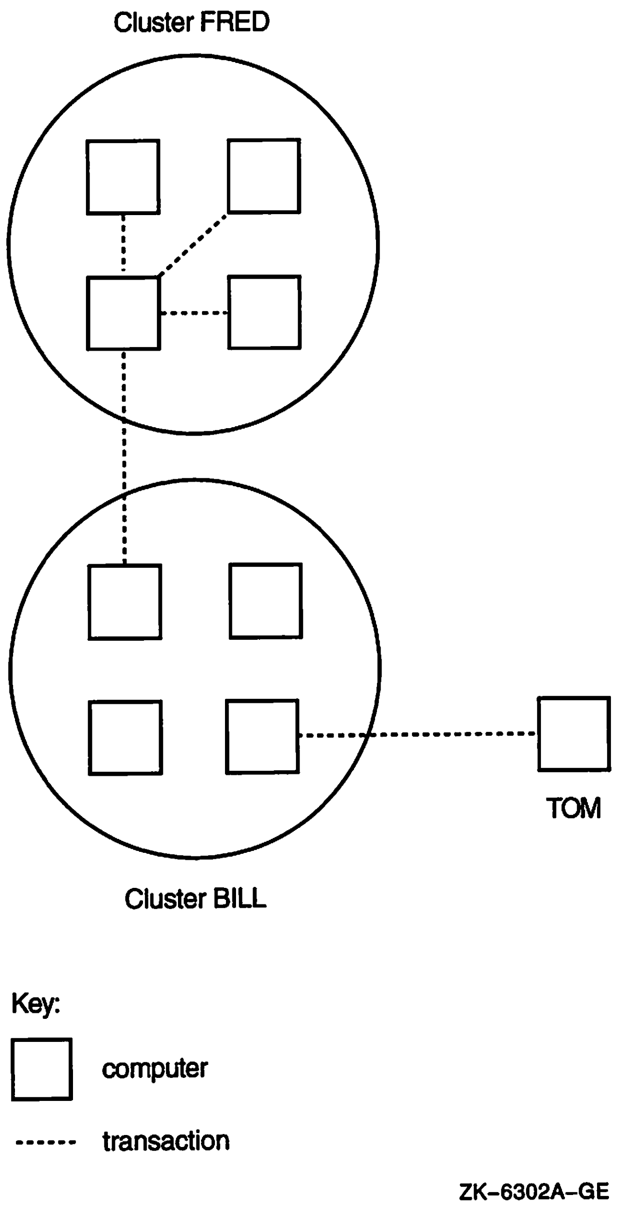
Make sure SCSNODE values are not the same as other computer names by following these steps.

Step	Action
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 involves computer C For an example, see Example of a Transaction Group.
2	For each computer in your transaction group: a. 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 b. 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.*

Example of a Transaction Group

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

Figure 3–1 Transaction Group



In this example:

- Transactions on a computer in cluster FRED involve other computers in cluster FRED, and a computer in cluster BILL. ctions 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

3.10 DECnet for OpenVMS AXP

The following sections describe new support and restrictions that apply to DECnet for OpenVMS AXP.

3.10.1 Cluster Alias Support Available

V1.5 DECnet cluster alias routing support is available in Version 1.5. Note, however, that the Product Authorization Key (PAK) name that enables DECnet for OpenVMS AXP cluster alias routing support (DVNETEXT) is different from the PAK name that enables OpenVMS VAX cluster alias routing support (DVNETRTG). The functions that are supported with the DVNETEXT (extended function) license differ from the VAX DVNETRTG license. DVNETEXT is supported only to enable Level 1 routing on AXP nodes acting as routers for a cluster alias.

In a dual-architecture VMScluster environment, the router for the cluster alias can be either an AXP system or a VAX system.

For additional information, see the chapter on networking management tasks in *A Comparison of System Management on OpenVMS AXP and OpenVMS VAX*.

3.10.2 Restrictions

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

- You *must* start the LAT software after you start DECnet. If you start DECnet after you start LAT, all existing LAT connections are terminated, and you might be unable to reconnect using LAT. This restriction only applies to Ethernet ports that will be running both LAT and DECnet (it does not apply to FDDI).
- Level 1 routing is available but is supported only on DECnet for OpenVMS AXP nodes acting as routers for a cluster alias. Routing between multiple circuits is not supported. Level 2 routing is not supported on DECnet for OpenVMS AXP nodes.
- Some line types are unsupported.

  OpenVMS AXP Version 1.5 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 AXP systems.

3.11 Reverting to Previous Device Protection Scheme

V6.1 Sites at which small numbers of unprivileged individuals must be able to allocate tape drives or disk drives for mounting private volumes may wish to add Access Control Lists to those devices and grant a general identifier to those individuals who are permitted to perform such allocation.

For sites where large numbers of unprivileged individuals must be able to allocate tape drives or disk drives, and where there is no concern about "denial of service" due to some user holding onto control of such a device, the following procedure can be used to revert to the previous device protection scheme on a global basis:

```
SYS$EXAMPLES:RESET_DEVICE_PROTECTION.COM
```

3.12 Unique Device Unit Numbers Required — Restriction

V1.5

In a dual-architecture cluster, all disk device unit numbers within an allocation class must be unique. For example, suppose an existing disk within the cluster is named $5$DUA10. When you add another disk device to the same allocation class (even if it is on another disk controller or another device type), the new disk device must be given a unit number other than 10.

3.13 Digital-Supplied Device Drivers — File Name Format Change

V1.0

Most device drivers supplied by Digital on AXP systems use the following format, where the *facility-name* is usually SYS:

facility-name$xxDRIVER.EXE

On VAX systems, no facility prefix is present or permitted for device drivers. They are simply named *xx*DRIVER.EXE.

On AXP systems, non Digital device drivers still can be called *xx*DRIVER.EXE.

3.14 Disk Defragmentation—Restriction

V6.1

Because the Movefile operation is not available on AXP systems, you must not use online disk defragmenters (such as DEC File Optimizer for OpenVMS) that rely on the Movefile operation.

Digital expects to remove this restriction in a future release.

3.15 Error Log Report Formatter (ERF)

The following sections describe features and restrictions that apply to ERF.

3.15.1 Features

V1.5

The Error Log Report Formatter (ERF) supports AXP error logs. VAX error logs are not supported. ERF is a native port of the VAX version.

Version 1.5 includes support for the FRU entry for the DEC 3000 Model 400 AXP computer.

3.15.2 Restrictions and Limitations

V1.0

The following restrictions apply to ERF:

- If, by mistake, a VAX error log is analyzed, the results would be unpredictable because VAX error logs are not supported.
- The qualifier /SUMMARY is not supported. Its use results in the following error message:

  ```
  %ERF-I-LOADER, error loading image, SYS$SYSTEM:ERFSUMM.EXE
  ```

- There is no Config Entry support for the DEC 7000 Model 600 AXP computer.

3.16 FDDI Clusters—Restriction Removed

V6.1

To enable FDDI clustering on OpenVMS AXP Version 1.5, you were required to set the system parameter PE3 to 1. For OpenVMS AXP Version T6.1-FT4, this requirement has been removed.

3.17 Help Message Utility (MSGHLP) Restriction

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 future releases.

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 may also be able to reuse some .MSGHLP files to insert comments to existing messages.

3.18 InfoServer Notes

The following notes pertain to the InfoServer.

3.18.1 Installing the InfoServer System on a DEC 4000 System

V6.1 If you use the InfoServer system to install the OpenVMS AXP Version T6.1-FT4 operating system on a DEC 4000 system, add the following item to the BOOT command:

```
-START 0
```

The following example shows a complete BOOT command:

```
>>> BOOT -FL 0,0 -FI APB_020 -START 0 lan-device-name
```

3.19 Updating InfoServer Software from ConDIST

V6.1 You can use Disc 1 of Digital's Consolidated Software Distribution (ConDIST) to update InfoServer software. To perform an update operation, log in to the InfoServer system and follow these steps:

1. Insert the disc 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:
   ```
   InfoServer> UPDATE SYSTEM DKn:
   ```

   On the InfoServer 1000 system:
   ```
   InfoServer> UPDATE SYSTEM DKn: FLASH
   ```

   This command moves the InfoServer software to the internal read/write device. The next time you boot the InfoServer system, it runs the updated software.

Note that because you cannot boot the server from the ConDIST disc, you should retain the InfoServer Software compact disc that shipped with your InfoServer Software kit.

3.20 Accessing ISO 9660 Compact Discs from an InfoServer System

V6.1 To make ISO 9660 compact discs available to OpenVMS client systems from an InfoServer system, the InfoServer manager must create services in the ODS_2 service class for these discs.

3.20.1 InfoServer Installation — Change in Service Name

V1.5 If you are using the InfoServer to install OpenVMS AXP Version 1.5, the default InfoServer service name displayed by the installation procedure is incorrect. The system prompts you to enter the InfoServer service name and displays the default name OPENVMSALPHA015. Do not use the default name.

Instead, enter OPENVMSAXP15 for the InfoServer service name, as shown in the sample display in the *OpenVMS AXP Version 1.5 Upgrade and Installation Manual.*

3.20.2 InfoServer Booting from DEC 7000 and DEC 10000 Systems

V6.1 InfoServer booting for operating system upgrades or installations on DEC 7000 and DEC 10000 systems is only supported with Version 3.0 of the InfoServer kernel.

A solution for this restriction will be provided in the future.

3.20.3 Losing Connection to the InfoServer

V6.1 If you booted the OpenVMS AXP T6.1-FT3 kit from an InfoServer, you might lose your connection to the InfoServer due to network hardware or software problems, or an overloaded network.

If you do lose the connection to the InfoServer is lost, the system will hang. To remedy the situation, try pressing Ctrl/Y. If Ctrl/Y fails to return you to the menu (which displays options for installing or upgrading, entering a DCL environment, or shutting down), then you have lost the connection to the distribution compact disc connected to the InfoServer.

To recover from this situation, do the following:

- If you had previously chosen the INITIALIZE option during the installation operation, reboot and perform the installation again.
- If you had previously chosen the PRESERVE option, first reboot and restore the backup of your target disk. Then perform the installation or upgrade again.

3.21 Installation and Upgrade

The following notes contain information on the installation or upgrade of OpenVMS AXP. <QAR epsilon_rel_notes 36, Larry Kilgallen; 2/2/94, John Knowles)

3.21.1 Error Message %RMS-E-DNF

V6.1 In certain circumstances it is possible for installation or upgrade to fail with the message:

```
Portion Done: 0%...10%...20%...30%...40%...50%...60%...70%...80%...90%
%PCSI-I-PRCOUTPUT, output from subprocess follows...
%RMS-E-DNF, directory not found

%PCSI-E-EXEFAIL, execute statement failed; status returned from DCL follows
-RMS-E-DNF, directory not found
```

The lack of further information is unfortunate, and changes will be made in the future to provide better feedback in situations of this nature.

In the meantime, this particular failure can be caused by a situation where the target disk for an installation or upgrade has at least one set of directories which looks like a system-specific root but is incomplete. An example would be:

```
DKA200:[SYS5]SYSCOMMON.DIR;1
DKA200:[SYS5.SYSCOMMON]SYSLIB.DIR;1
DKA200:[SYS5.SYSCOMMON.SYSLIB]MY_FILE.EXE;1
```

Restoring the backup made of your target disk before you started the installation or upgrade and removing or renaming the partial root will allow a successful installation or upgrade.

3.21.2 VIOC Cache Size Reduced

V6.1 When booting from the Installation CD–ROM, managers of systems that have only 32 megabytes of memory will see the message:

```
VIOC Cache size reduced
```

Digital intends to correct the slight installation-time performance degradation indicated by this message for a subsequent release.

3.22 Removing the OpenVMS AXP Operating System

V6.1 Although the use of the PRODUCT REMOVE command is not officially supported in OpenVMS AXP Version T6.1-FT4 for the removal of the OpenVMS operating system, it is possible to use the PRODUCT REMOVE command to remove most of the OpenVMS AXP operating system from a system disk, without effecting user files on the disk.

Follow these steps to remove OpenVMS AXP:

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. Shutdown and and boot from the distribution CD or from a system disk other than the one from which OpenVMS AXP is being removed. Execute the following DCL commands; substitute the device name of the disk from which OpenVMS AXP 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$DESTINATION <target-disk>:[VMS$COMMON]
   $ DEFINE/NOLOG PCSI$DSPECIFIC <target-disk>:[SYSx.]
   $ PRODUCT REMOVE VMS /REMOTE
   ```

   If you are not running from the distribution CD, you will need to be logged in to a privileged account.
3. After the remove operation completes, you should review the target-disk to determine if you wish 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 you on the target disk:

     ALPHAVMSSYS.PAR;1
     MODPARAMS.DAT;1
     PAGEFILE.SYS;1
     SWAPFILE.SYS;1

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

  LMF$LICENSE.LDB;1
  PCSI$FILE_SYSTEM.PCSI$DATABASE;1
  PCSI$PROCESSOR.PCSI$DATABASE;1
  PCSI$ROOT.PCSI$DATABASE;1

  NOTE: Do not remove the *.PCSI$DATABSE files if you have layered products installed on this disk, or if you want to maintain a history of software installation on this disk.
  RIGHTSLIST.DAT;1

4. You should also review the target-disk for the directory structures [VMS$COMMON...] and [SYSx...], which will remain after removing OpenVMS. You may wish to remove these directories.

3.22.1 Remove DECnet OSI

V6.1

If you have installed version T2.1-FT3 of DECnet OSI (also called DECnet Phase V) you must remove it before upgding to OpenVMS T6.1-FT4. The OpenVMS installation procedure will remind you of this if you do not remove DECnet OSI.

During the removal of DECnet OSI you will see four PCSI-E-RESGEN errors. The POLYCENTER Software Installation utility will recommend terminating for each of these errors. However, the DECnet OSI removal procedures will "fix up" this problem. Therefore, yo should answer "NO" when you are asked "Do you want to terminate?" after each or these four errors. (Note that the default answer is "YES", do terminate; you must explicitly type "NO".)

The removal dialog for DECnet OSI may look like the follwing:

```
$ product remove decnet_osi

The following product has been selected:
DEC AXPVMS DECNET_OSI T2.1-FT3

Do you want to continue? [YES]
The following product will be removed:
DEC AXPVMS DECNET_OSI T2.1-FT3

%PCSI-E-RESGEN, the file [SYS$LDR]NET$CSMACD.EXE provided by
DEC AXPVMS DECNET_OSI T2.1-FT3 will remain installed;
replacement material unavailable
Terminating is strongly recommended.  Do you want to terminate? [YES] NO

%PCSI-E-RESGEN, the file [SYS$LDR]NET$FDDI.EXE provided by
DEC AXPVMS DECNET_OSI T2.1-FT3 will remain installed;
replacement material unavailable
Terminating is strongly recommended.  Do you want to terminate? [YES] NO

%PCSI-E-RESGEN, the file [SYS$LDR]SYS$NETWORK_SERVICES.EXE provided by
DEC AXPVMS DECNET_OSI T2.1-FT3 will remain installed;
replacement material unavailable
Terminating is strongly recommended.  Do you want to terminate? [YES] NO

%PCSI-E-RESGEN, the file [SYSEXE]NCP.EXE provided by
DEC AXPVMS DECNET_OSI T2.1-FT3 will remain installed;
replacement material unavailable
Terminating is strongly recommended.  Do you want to terminate? [YES] NO
....
```

3.22.2 Spurious Message during Booting

V6.1

You may see the following message on some systems during boot. Do not be alarmed; contrary to the message, the installation has completed correctly.

```
%INSTALL-E-RESFAIL, failed to install image with /RESIDENT qualifier
-INSTALL-E-NOGHREG, insufficient memory in the code granularity hint
region
%INSTALL-I-NONRES, installed image non-resident with other specified
options
```

You can either ignore this message or change the value of the SYSGEN parameter ITB_ENTRIES from 1 to 2, which eliminates this error message during the next reboot.

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

3.22.3 Volume Shadowing

V6.1

Please refer to the Volume Shadowing documentation for information about Volume Shadowing.

Caution

Do *not* attempt to install the Volume Shadowing layered product on OpenVMS AXP Version T6.1-FT4. The resulting system will not work. An OpenVMS AXP Version 1.5 system on which the Volume Shadowing layered product was installed can be upgraded to OpenVMS AXP Version T6.1-FT4.

As part of upgrading a shadowed system disk to OpenVMS AXP Version T6.1-FT4, the shadow set must be temporarily broken. See the *OpenVMS AXP Version 6.1 Upgrade and Installation Manual* for more information.

3.22.4 No Upgrade from OpenVMS AXP Version 1.0

V6.1

An upgrade from OpenVMS AXP Version 1.0 is not supported and does not work. An upgrade from field-test versions of OpenVMS AXP Version 1.5 is not supported.

3.23 Changes to the Install Utility

V1.5

In Version 1.0, you could use the Install utility to install shareable images resident. Version 1.5 lets you install main images resident. Note that these main images must be linked with the /SECTION_BINDING qualifier before they can be installed resident.

An attempt to install an image /RESIDENT may fail if the code granularity hint region contains insufficient memory or if the image was linked incorrectly (that is, linked without the /SECTION_BINDING=CODE qualifier). If the attempt fails for either of these reasons, INSTALL now automatically attempts to install the image without the /RESIDENT qualifer but with all other specified options.

When this occurs, the Install utility displays messages, as shown in the following example:

```
$ INSTALL
INSTALL> replace/resident sys$library:foo.exe
%INSTALL-E-RESFAIL, failed to install image with /RESIDENT qualifier
-INSTALL-E-NOGHREG, insufficient memory in the code granularity hint region
%INSTALL-I-NONRES, installed image non-resident with other specified options

$ INSTALL
INSTALL> create/resident sys$library:foo2.exe
%INSTALL-E-RESFAIL, failed to install image with /RESIDENT qualifier
-INSTALL-E-NOGHREG, insufficient memory in the code granularity hint region
%INSTALL-I-NONRES, installed image non-resident with other specified options
```

3.24 I/O Buffer Cache Pools—Changes

V6.1 The minimum sizes of three of the I/O buffer cache pools have been increased:

Pool	Parameter	Old Minimum	New Minimum
Directory pool	ACP_DIRCACHE	2 blocks	4 blocks
Header pool	ACP_HDRCACHE	3 blocks	8 blocks
Bitmap pool	ACP_MAPCACHE	1 blocks	2 blocks

3.25 LAT Notes

This section contains information about LAT software.

3.25.1 Starting LAT Software

V1.0 OpenVMS AXP supports the LAT software when running on a suitably configured system. See the *OpenVMS System Manager's Manual* for more information about starting and using the LAT software.

Note

You *must* start the LAT software after you start DECnet. If you start DECnet after you start LAT, all existing LAT connections are terminated, and you might be unable to reconnect using LAT. This restriction applies only to Ethernet ports that will be running both LAT and DECnet (it does not apply to FDDI).

Always start LAT from the SYSTEM account because this account typically has appropriate privileges and quotas.

3.25.2 Change in LAT Behavior

V1.5 After you enter the LATCP CREATE LINK /DECNET command, the following message may be displayed:

```
%SYSTEM-F-BADPARAM, bad parameter value
```

This may indicate that the SCSSYSTEMID system parameter is set to an illegal value. Use the following formula when setting this parameter:

```
(1024 * a) + n
```

In the formula, *a* is the DECnet area, and *n* is the DECnet node number.

When you use the /DECNET qualifier with the LATCP CREATE LINK command, be sure that the value of the SCSSYSTEMID system parameter is within the range of 1025 to 65535, as determined by the formula above. The value for this parameter should match the results obtained by using the formula. If the value is outside the range of 1025 to 65535, the LAT protocol is unable to start.

When you use the /NODECNET qualifier, the LAN device driver code decides what address to use. The decision is based on the following criteria:

- If SCSSYSTEMID is set to 0 but DECnet is already running on an Ethernet controller, the LAN device code allows the LAT software to use the same address as DECnet (AA-00-04-00-xx-xx).
- If SCSSYSTEMID is set to 0 and DECnet is not running, the 08-00-2B-xx-xx-xx address is used.
- If the setting for SCSSYSTEMID is the same as the DECnet node number and DECnet is not running, the LAN device code forces LAT to use the AA-00-04-00-xx-xx address.

Restriction

If DECnet is configured on the system (or if the system is part of a cluster), SCSSYSTEMID may contain a nonzero value. Normally this is not a problem unless the system has two or more LAN controllers connected to the same logical LAN.

For example, if your system has an FDDI controller and an Ethernet controller, your site may be configured so that the FDDI ring attached to the FDDI controller and the Ethernet segment attached to the Ethernet controller are bridged by a 10/100 LAN bridge (FDDI-to-Ethernet). In this configuration, it is impossible to run LAT over both controllers.

In such a configuration, you *must* run LAT and DECnet over the same controller if SCSSYSTEMID is not 0. If you fail to do so, DECnet starts first, which in turn causes the LAT startup on the other controller to fail. This failure occurs because LAT startup tries to use the AA-00-04-00-xx-xx address (the DECnet LAN address); however, because DECnet is already using this address on another controller, the data link layer prevents the LAT startup from using that address. (In a single logical LAN, all data link addresses must be unique. Because both controllers try to use the same address, it is no longer unique.)

Using the following command to create the LAT link also fails because the LAN driver tries to use the address based on SCSSYSTEMID:

```
LATCP> CREATE LINK LAT$LINK_2 /NODECNET
```

If SCSSYSTEMID is set to 0, configuring LAT and DECnet on different controllers is possible. However, in a cluster environment, SCSSYSTEMID cannot be set to 0.

3.25.3 LAT Restrictions and Known Problems

3.25.3.1 New Baud Rate Support for VT500 Series Terminals

V6.1

VT500 series terminals are now supported with the following baud rates:

- 57600 baud
- 76800 baud
- 115200 baud

The only terminal controller capable of supporting all these speeds is a DECserver 700. However, the DECserver 90TL and DECserver 90M support terminal port speeds up through 57600. When the DECserver communicates with the OpenVMS AXP system using the LAT software, a restriction allows the LAT software to report speeds only up to 57600.

This restriction will be addressed in a future release.

3.25.3.2 BYTLM Quota and LTA Devices

V6.1 Previously, if you created an LTA device and assigned a logical name to it without first specifying the device number, the system consumed the process BYTLM; bytes would only be credited back to the creating process when the LTA device was deleted. If several LTA devices were created in the previously described manner, system startup processes ran out of BYTLM and would not finish.

Beginning with Version Version T6.1-FT4 of the OpenVMS AXP operating system, LTDRIVER credits the byte count of a created LTA device back to the creating process if the device is going to be a permanent LTA device (for example, APPLICATION, DEDICATED, or LIMITED).

3.26 License Management Facility (LMF)

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

3.26.1 PAKs on AXP and VAX Systems

V1.0 Availability Product Authorization Keys (PAKs) are available for OpenVMS AXP. An OpenVMS AXP PAK is identified by the keyword ALPHA in the PAK's option field.

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

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

Caution

By default, all AXP availability PAKs look disabled to a VAX system. Never use the DELETE/STATUS=DISABLED command from a VAX system on an LDB that contains AXP PAKs. If you do, all AXP PAKs will be deleted.

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

3.26.2 LICENSE Command Restrictions Removed

V6.1

In previous versions of OpenVMS, from a VAX system you could not use the following LICENSE commands on a PAK containing the ALPHA option:

- COPY
- DELETE/STATUS
- DISABLE
- ENABLE
- ISSUE
- LIST
- MOVE
- REGISTER

This restriction is lifted in OpenVMS VAX Version 6.1 and OpenVMS AXP Version Version T6.1-FT4.

3.27 Monitor Utility

3.27.1 Restrictions

V1.5

The Monitor utility (MONITOR) is nearly identical to that found on VAX systems. However, you should read *A Comparison of System Management on OpenVMS AXP and OpenVMS VAX* for information about the differences between MONITOR for AXP systems and MONITOR for VAX systems. One significant difference is that on AXP systems, the POOL class does not exist. Instead, adaptive pool management is used to enhance system performance and reduce overall pool memory requirements. See *A Comparison of System Management on OpenVMS AXP and OpenVMS VAX* for information about adaptive pool management.

Also refer to the *OpenVMS System Manager's Manual* for more information about the Monitor utility.

3.27.2 Remote Monitoring Restriction in Mixed-Version VMSclusters

V1.5

Restriction:

Remote monitoring is a feature of the Monitor utility that allows you to perform live monitoring of any node in a VMScluster. It is accomplished either by issuing the MONITOR CLUSTER command, or by adding the /NODE qualifier to any live MONITOR request.

Due to an incompatibility in the definition of MONITOR's collected data, it is not possible to perform remote monitoring across nodes running different versions of OpenVMS.

In a mixed-version VMScluster, remote monitoring is allowed only between nodes running the same OpenVMS version. If an attempt is made to monitor a remote node of a different version, the following message will be displayed:

```
%MONITOR-E-SRVMISMATCH, MONITOR server on remote node is an incompatible
version
```

Workaround:

You can, however, obtain MONITOR data from a remote node of a different version of OpenVMS by recording the data on the remote node, and then using the MONITOR playback feature to examine it on the local node. Please see the *OpenVMS System Management Utilities Reference Manual* for details on using the MONITOR recording and playback features.

3.27.3 MONITOR RMS—Items Implemented

V6.1

Two data items under the RMS locking statistics section of the MONITOR RMS command are implemented in this release:

- Bucket splits (bucket splits involving one new bucket)
- Multi-bucket splits (bucket splits involving more than one new bucket.

3.28 Mount Utility

V1.5

In this version of the operating system, the MOUNT command is supported for tape and disk volumes. The MOUNT/CLUSTER command is also supported.

3.28.1 Foreign Mounted Tape: Maximum Record-Size Restriction Removed

V6.1

The maximum record size for a variable record format for a sequential file on a foreign mounted tape is no longer restricted to the ANSI tape maximum of 9995. The same maximum record sizes that apply to a sequential file on disk now apply to a sequential file on a foreign mounted tape.

3.28.2 PATHWORKS Access to ISO 9660 in a WAN/LAN Environment

V1.5

To access ISO 9660 volumes in a WAN/LAN environment, issue a MOUNT /SYSTEM command for the ISO 9660 volume.

On the PC client node, assign the volume to a PC device using the appropriate command. For example, in an MS–DOS environment, the assignment command might look like the following:

```
B:> USE ?: \\MYNODE\DISK$VOLUME:[000000]%VMSUSER *
```

The assignment command has the following format:

- *USE ?:* commands the USE utility to assign the next available PC device.
- \\MYNODE indicates the OpenVMS node.
- \DISK$VOLUME:[000000] indicates the volume and directory.
- *%VMSUSER* * indicates the access control string. The asterisk in the access control string causes MS–DOS to prompt you for the password.

3.28.3 Extraneous Mount Count Dismounting from Shared Mount

V6.1

If a user has done a MOUNT/SHARE on the system disk, a subsequent dismount may produce the message:

```
%DISM-W-CANNOTDMT, AXP27$DKA300: cannot be dismounted
%DISM-W-USERFILES, 1 user file open on volume
```

This happens because the program DISMOUNT.EXE on the system disk is opened by the user at the time the dismount is attempted.

MOUNT/SHARE of disks already mounted /SYSTEM is used to retain a lock on disk availability even if the disk should be dismounted on a systemwide basis. This practice is not required in the case of the system disk, but it can occur as a result of invoking a general purpose command procedure which is sometimes used on the system disk and sometimes used on other disks.

As a workaround, you can INSTALL the DISMOUNT.EXE image, which keeps it from showing up as an open file and thus prevents the message.

3.28.4 ISO 9660 Support in Mixed-Architecture VMScluster Systems

V6.1

On OpenVMS systems, do not specify the /CLUSTER qualifier when mounting ISO 9660-formatted CD–ROMs in a VMScluster 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.

3.29 Password Generation

This section contains information about generating passwords.

3.29.1 Password Generation Algorithm—Restriction

V1.0

On AXP systems, the password generation algorithm allows for future use of non-English password generation databases. Because of this, the password generation logic does not return information necessary to perform English word hyphenation. As a result, the SET PASSWORD command cannot display a hyphenated word list, as it does on VAX systems. This is a permanent change in behavior on AXP systems.

3.29.2 SET PASSWORD/GENERATE Command

V6.1

When using the factory installed version of OpenVMS AXP, the command SET PASSWORD/GENERATE may output blank lines instead of generated passwords. This is caused by an incorrect file protection on the file SYS$SHARE:VMS$PASSWORD_DICTIONARY.DATA. The protection on this file must be WORLD:READ and may be corrected with the command $ SET PROTECTION=WORLD=READ SYS$SHARE:VMS$PASSWORD_DICTIONARY.DATA

3.30 Performance Degradation Due to Insufficient Swap File Space

V1.5

Performance may degrade on systems using large working sets if the swap file space is insufficient.

Configurations with large process working sets that run under heavy load conditions must have adequate swap file space. Processes can be swapped out with a working set size up to the value of WSQUOTA. The I/O that is issued to swap out the process can be as large as the setting for WSQUOTA and requires this much space in the swap file. (WSQUOTA is a process parameter that you can adjust using the Authorize utility.)

To provide adequate swap file space, periodically run AUTOGEN with the FEEDBACK option. When running AUTOGEN, make sure that:

- The system has been up long enough to have run a typical load (at least 24 hours).

- There is no hard-coded SWAPFILE value in the SYS$SYSTEM:MODPARAMS.DAT file. (A hard-coded SWAPFILE value prevents AUTOGEN from sizing the swap files correctly.)

AUTOGEN resizes swap files provided there is adequate space on the volumes containing the swap files. If there is insufficient space, AUTOGEN recommends swap file sizes. Use swap file sizes at least as large as those recommended by AUTOGEN. If the swap file sizes need to be increased and insufficient free space exists on the volume containing the swap files, perform one of the following steps after AUTOGEN finishes:

- Free up space on the volume, and increase the swap file size to at least the value recommended by AUTOGEN.
- Move the swap files to a volume with sufficient space. Make sure that the swap files are at least as large as those recommended by AUTOGEN.
- Add another swap file. The total space in all swap files must be at least as large as the total size recommended by AUTOGEN.

3.31 PMAZB Adapter — Limited Support Available

V1.5

The *OpenVMS AXP Version 1.5 Release Notes* manual and the *OpenVMS AXP Version 1.5 Upgrade and Installation Manual* state that the TURBOchannel PMAZB adapter is supported in OpenVMS AXP Version 1.5.

The PMAZB adapter is supported in OpenVMS AXP Version 1.5 for data devices only. No support is provided for OpenVMS AXP Version 1.5 system disks that are connected to PMAZB adapters.

3.32 RZ24L 245MB SCSI Fixed-Disk Drive Supported

V1.5

The RZ24L 245MB SCSI fixed-disk drive is fully supported on DEC 3000 Model 300 and Model 300L systems running OpenVMS AXP Version 1.5.

3.33 Privileges, Protections, and Quotas

V1.0

The default values for a number of process limits and quotas are higher than those on VAX systems. Refer to the chapter on system setup tasks in *A Comparison of System Management on OpenVMS AXP and OpenVMS VAX* for more information about default values for process limits and quotas.

3.34 Security Auditing

3.34.1 Audit Server Database Name Change

V6.1

The security auditing server process maintains the set of auditing events that are enabled on the system. The audit server process stores this information in a file in SYS$MANAGER. In OpenVMS AXP Version Version T6.1-FT4, the name of this file has been changed from AUDIT_SERVER.DAT to VMS$AUDIT_SERVER.DAT because the format of the V6.1 file differs from earlier versions.

3.34.2 Audit Server Database Files in Mixed-Version Clusters

V6.1 As Section 3.34.1 describes, the name for the audit server database changes in Version Version T6.1-FT4 from AUDIT_SERVER.DAT to VMS$AUDIT_SERVER. When redirecting this file off the system disk, system managers doing a rolling upgrade must provide logicals for both AUDIT_SERVER and VMS$AUDIT_SERVER in SYSECURITY.COM. As before, these logical names must be defined /SYSTEM/EXEC.

3.34.3 Audit Log Files in Mixed-Version Clusters

V6.1 The Audit Analysis utility (ANALYZE/AUDIT) running on Version 1.5 systems is unable to process Version Version T6.1-FT4 audit log files. You must use Version Version T6.1-FT4 of ANALYZE/AUDIT to process Version Version T6.1-FT4 audit log files. The recommended procedure is to maintain separate audit log files on mixed-version clusters.

If redirecting the audit log files, the command SET AUDIT/JOURNAL /DESTINATION=filespec should be issued on both a Version 1.5 node and on a Version Version T6.1-FT4 node. The destination filespec is stored in the audit server database file. By default, the files are stored in SYS$COMMON:[SYSMGR] and are called SECURITY_AUDIT.AUDIT$JOURNAL and SECURITY.AUDIT$JOURNAL respectively. See the *OpenVMS VAX Guide to System Security* for further information.

3.34.4 Auditing Configuration

V6.1 In Version 1.5, SET AUDIT commands in different command procedures or startup files established audit server settings. With Version Version T6.1-FT4, the audit server stores auditing settings in a permanent database. The settings in the database VMS$AUDIT_SERVER.DAT are reinstated on each system startup.

Therefore, system managers running V6.1 systems can remove SET AUDIT commands from startup files and command procedures. After reviewing the new auditing system described in *OpenVMS VAX Guide to System Security*, managers can configure a Version Version T6.1-FT4 auditing system. Section 3.35.1 provides a brief summary of the outstanding changes.

3.35 Security Changes

This section contains a summary of all the security changes introduced in OpenVMS AXP Version Version T6.1-FT4, followed by specific notes related to security.

3.35.1 Summary of Changes

V6.1 OpenVMS AXP Version Version T6.1-FT4 offers significant enhancements to system security. Sites interested in implementing the new security features can find a comprehensive overview of changes in Chapter 1. Whether you choose to implement the features, note the following changes because they can impact your daily operations.

- The auditing subsystem is different from what existed in earlier versions. Chapter 6 of the *OpenVMS VAX Guide to System Security* explains how auditing works in Version Version T6.1-FT4. Note the following changes, in particular:
  - Auditing characteristics are now permanent and distributed clusterwide.

  - Alarm ACEs no longer send event messages to the system audit log file. For a permanent record of events, use Audit ACEs to write access events to the system audit log file.
  - Alarm message text is not written to the operator log file (OPERATOR.LOG).
  - The operating system does not allow logins if it is unable to open the system audit log file.
  - When you redirect the audit log file, the change takes place immediately. There is no longer a need to enter the DCL command SET AUDIT/NEW_ LOG.
  - The audit server now monitors the audit log file to determine how much disk space is available for audit event messages, and when space runs low, it pre-allocates blocks (Section 3.35.9).
- Captive and restricted accounts have changes in spawn operations:
  - By default, captive accounts cannot use the DCL command SPAWN (Section 3.35.5).
  - The MAIL utility and DECTPU now allow the SPAWN comand in restricted accounts (Section 3.35.5).
- With access control entries (ACEs) carrying the Hidden, Nopropagate, or Protected attribute, there is a change in the way DCL commands perform copy operations (Section 3.35.19).
- The default protection for disk and tapes has changed (Section 3.35.8).
- OpenVMS AXP now performs a read and write access check on unshared devices ($QIO) (Section 4.8.3, *OpenVMS VAX Guide to System Security*).
- Terminal ownership is changed at login from the value in the template to the UIC of the process logging in (Section 4.8.3.5, *OpenVMS VAX Guide to System Security*). This change is necessary because in Version Version T6.1-FT4, the operating system checks for read and write access to a terminal.
- Objects with a UIC of [0,0], such as devices, no longer have control access by default (Section 3.35.2).
- The READALL privilege no longer grants control access (Appendix A, *OpenVMS VAX Guide to System Security*).
- With terminal devices, default protection and ownership are no longer set by the system parameters TTY_DEFPROT and TTY_OWNER. The operating system now uses the security profile template DEVICE.TERMINAL to set the default protection and owner (Section 3.35.17 in this book and Section 4.8.3, *OpenVMS VAX Guide to System Security*).
- The $ASSIGN system service once required read access to a device to assign a channel, but now the system service permits assignment if a process has read, write, or control access (*OpenVMS System Services Reference Manual*).
- Prior to Version Version T6.1-FT4, the security characteristics for volumes and devices were combined but now volumes and devices have separate UICs, protection codes, and ACLs (Section 3.35.16 in this book and the *OpenVMS VAX Guide to System Security*.)

- Batch and print queues have different access requirements:
  - Prior to Version Version T6.1-FT4, submitting a job to a queue required read access and submitting a job with the /DELETE qualifier required delete access. Now, additional checks are performed when the job executes.
  - Displaying queue security information now requires read access to the queue.
  - Read and delete access to queues differs, depending on whether access is granted through a protection code or an ACL (Section 4.8.7, *OpenVMS VAX Guide to System Security*).
- When you create a new version of a file, the file system now checks for write access to the previous file version. Although the old version is not modified, superceding the file effectively alters the data that existed.
- The rights database is no longer world-readable (Chapter 1).
- The system services $FIND_HELD, $ASCTOID, and $IDTOASC now check the requestor's access to an identifier under certain conditions (Section 5.4.2, *OpenVMS VAX Guide to System Security*).

3.35.2 Access Restriction to Objects with UIC [0,0]

V6.1

The Check Access ($CHECK_ACCESS) and Check Protection ($CHKPRO) system services now deny control access to objects with the user identification code (UIC) [0,0].

In previous versions of the operating system, the $CHKPRO and $CHECK_ACCESS system services granted full access to objects with an owner UIC of [0,0] and an empty access control list (ACL). Granting full access did not compromise system security because, in general, the only objects owned by [0,0] were devices. To change the security profile for a device, you had to have privileges. (The SET PROTECTION/DEVICE command requires OPER privilege, and the SET ACL/CLASS=DEVICE command requires SYSPRV or BYPASS privilege).

The system now uses the SET SECURITY and SHOW SECURITY commands (which invoke the $CHECK_ACCESS and $CHKPRO system services) to manipulate the security profile for all types of objects consistently. If you have control access to an object, you can change the security profile for the object. Therefore, control access to objects with a UIC of [0,0] is no longer granted by default.

3.35.3 Alarm Access Control Entries—No Permanent Record

V6.1

Prior to OpenVMS AXP Version Version T6.1-FT4, an Alarm_Journal access control entry (ACE) reported an alarm to security operator terminals and to the audit log file. The Alarm_Journal ACE has been renamed Alarm ACE and now generates the alarm only.

To obtain a permanent record, add an Audit ACE to each file that currently has an Alarm ACE; or, replace the Alarm ACE on each file with an Audit ACE.

3.35.4 ANALYZE/AUDIT in a Mixed-Version Cluster

V6.1

In a mixed-version cluster, an audit log file contains entries from systems running different versions of the operating system. To analyze the log file, you must invoke the Audit Analysis (ANALYZE/AUDIT) utility from a node running Version Version T6.1-FT4.

If you invoke the ANALYZE/AUDIT utility from a node running an earlier version of OpenVMS AXP, it fails.

3.35.5 Captive and Restricted Accounts—SPAWN and LIB$SPAWN Changes

V6.1

Beginning with OpenVMS AXP Version Version T6.1-FT4, restricted accounts can use the DCL command SPAWN within the MAIL utility or DECTPU. This change makes it possible to use the RESTRICTED flag to force additional authentication or authorization checks within a login command procedure (or SYLOGIN) without otherwise affecting a user's environment. Restricted accounts now behave like regular accounts once the initial login sequence has completed.

Also, OpenVMS AXP Version Version T6.1-FT4 prohibits all uses of the SPAWN command or the RTL routine LIB$SPAWN within captive accounts unless it is defined as trusted. A new /TRUSTED qualifier can be used to connote spawn requests that are trusted to perform their function in a secure manner. A new flag bit (%X40) has been defined for the FLAGS argument to the LIB$SPAWN routine. This change was made in response to a large number of problem reports having to do with SPAWN commands originating in captive command procedures (for example, "SPAWN @TT:").

To preserve compatibility with existing applications that run in captive accounts and contain LIB$SPAWN calls, a flag bit (%X40) in the new SECURITY_POLICY system parameter can be set to allow captive accounts to continue to use SPAWN commands and LIB$SPAWN calls. Assuming the default value for the SECURITY_POLICY parameter (7), use the following commands to set this flag:

```
$ RUN SYS$SYSTEM:SYSMAN
SYSMAN> PARAM USE CURRENT
SYSMAN> PARAM SET SECURITY_POLICY 71 ! 64 + 7
SYSMAN> PARAM WRITE CURRENT
```

3.35.6 Disk and Tapes—Access Control Lists Restored After System Reboots

V6.1

For disk and tape devices that are available clusterwide, OpenVMS AXP now stores security profiles in a permanent database. The system resets the owner, protection code, and ACL automatically for a device when the system reboots. You do not need to execute SET ACL (or SET SECURITY) commands in your system startup file to restore ACLs.

Note

If you have a mixed-version cluster, you must manage the different nodes in the cluster according to the version of OpenVMS AXP running on the node, as follows:

- On systems running versions of OpenVMS AXP to Version Version T6.1-FT4—The system startup files must include SET ACL commands to restore ACLs on disk devices.

- On systems running Version Version T6.1-FT4—The system startup files must not include SET ACL commands, because the SET ACL command changes the device ACL (not the volume ACL). For more information about device and volume ACLs, see Section 3.35.16.

3.35.7 Reverting to Previous Device Protection Scheme

V6.1

Sites at which small numbers of unprivileged individuals must be able to allocate tape drives or disk drives for mounting private volumes may wish to add Access Control Lists to those devices and grant a general identifier to those individuals who are permitted to perform such allocation.

For sites where large numbers of unprivileged individuals must be able to allocate tape drives or disk drives, and where there is no concern about "denial of service" due to some user holding onto control of such a device, the procedure:

```
SYS$EXAMPLES:RESET_DEVICE_PROTECTION.COM
```

can be used to revert to the previous device protection scheme on a global basis.

3.35.8 Disk or Tape Devices—Default Access

V6.1

In previous versions of OpenVMS AXP, a disk or tape device had an owner UIC of [0,0] and the following default protection code:

```
System:RWED, Owner:RWED, Group:RWED, World:RWED
```

In previous versions, this type of protection allowed any user to access the device. When a user dismounted a volume, the system reset the device protection to these default values.

In OpenVMS AXP Version Version T6.1-FT4, the default protection for disk or tape devices is as follows:

- The owner UIC is [SYSTEM]
- The UIC-based protection code is (System:RWLP, Owner:RWLP, Group:R, World:)

When applied to shareable devices, access types have the following definitions:

Access Type	Definition
Read	The right to issue requests to the device
Write	The right to issue write requests to the device
Logical I/O	The right to issue logical I/O requests to the device
Physical I/O	The right to issue physical I/O requests to the device

To initialize or mount a volume on a device, you must have read, write, or control access to the device. (There are also some additional requirements for initializing a volume that are unrelated to device protection.) System managers (users in the system category) and users with the privilege SYSPRV have this type of access.

The system manager, who has control access, can change the protection. For example, to change the protection code for the device DUA0, the system manager can enter the following command:

```
$  SET SECURITY/PROTECTION=(S:RWLP,O:RWLP,G:RW,W:RW)/CLASS=DEVICE DUA0:
```

This command changes the current protection code to allow any user to initialize or mount a volume on the device DUA0. Access to the device is then similar to the access allowed in previous versions of the operating system.

Note

Once a volume is mounted, the device protection has no effect on file operations on the mounted volume.

3.35.9 Disk Space for Audit Log File Now Pre-Allocated

V6.1 The audit server pre-allocates disk space to the security audit log file to ensure there is adequate space for event messages. Whenever the file runs low on available blocks, the audit server extends the audit log file. A system manager can use the DCL command DIRECTORY/SIZE/ALL to determine how much of the allocated disk space is being used.

3.35.10 Logical Name Table Ownership

V6.1 OpenVMS Version Version T6.1-FT4 changes the default owner of a group logical name table. In earlier releases, the default owner was [<group>,0]. The default is [<group>,*].

Because the operating system takes the user number from the template profile LOGICAL_NAME_TABLE.GROUP, you can configure a system where group tables are owned by [<group>,0] by modifying the template, as follows:

```
$ SET SECURITY/CLASS=SECURITY_CLASS/PROFILE=TEMPLATE=GROUP -
_$ LOGICAL_NAME_TABLE /OWNER=[0,0]
```

3.35.11 Password History Changes

This section contains changes to password history.

3.35.11.1 History Records Removed on Deletion of User Account

V6.1 Prior to OpenVMS AXP Version Version T6.1-FT4, password history records associated with a user account were not removed from the password history database when the user authorization record was removed from the system user authorization file (SYSUAF). Starting with OpenVMS AXP Version Version T6.1-FT4, associated password history records are removed from the password history database whenever a user account is deleted.

3.35.11.2 Passwords Older than Lifetime Now Valid

V6.1 The OpenVMS password history database maintains a history of previous passwords associated with each user account. By default, the system retains these history records for one year. Prior to OpenVMS AXP Version Version T6.1-FT4, old history records were removed only after a successful password change. This means that a user was unable to change a password to one that existed in the password history database even when the matching entry was older than the system password history lifetime.

At the request of several customers, this behavior has been changed. Password history records that are older than the password history lifetime are no longer considered valid and are allowed as a valid password choice.

3.35.12 PRINT/USER Command Requires Legitimate Access

V6.1

To meet C2/B1 security requirements, the queuing system performs additional access checking, to make sure users have legitimate access to files being printed.

In earlier versions of the operating system, the PRINT/USER command allowed you to print a file for a user who otherwise would not have access to the file. Similarly, the PRINT/USER/DELETE command allowed you to have a file deleted on behalf of a user who did not have delete access to the file.

In OpenVMS AXP Version Version T6.1-FT4, the user specified in the /USER qualifier must have read access to the file being printed. In addition, the user must have delete access to any file printed with the PRINT/USER/DELETE command.

For more information on the /USER and /DELETE qualifiers of the PRINT command, see the *OpenVMS DCL Dictionary*.

3.35.13 Privilege Auditing in a Mixed-Version Cluster

V6.1

OpenVMS AXP Version Version T6.1-FT4 allows you to audit all use of privilege by operators and system administrators. In a mixed-version cluster, you can enable privilege auditing only on those nodes running Version Version T6.1-FT4.

For nodes running Version Version T6.1-FT4, privilege auditing does not report the use of privilege in privileged, installed images. For example, the system does not record a privilege audit event when someone enters the SHOW USERS command (which invokes an image installed with the WORLD privilege). This audit filtering both reduces the amount of overhead required for privilege auditing and ensures that the system audits only the actual use of privilege by privileged processes.

In a mixed version cluster, when a request is initiated from within a privileged image on a node running an earlier version of software, the Version Version T6.1-FT4 system cannot detect that the remote image was installed. Therefore, the system always generates a privilege audit event. This explains why privilege audit events instigated by images installed with privilege can appear in the audit log file.

3.35.14 ODS-2 Disk Volumes Are Fully Supported Protected Objects

V6.1

You can now use ACLs to protect ODS-2 disk volumes. The entire security profile (owner UIC, protection code, and ACL) is stored on the volume. If you change the volume security profile for a volume mounted clusterwide, the change is distributed to all the nodes in the cluster. If you dismount and remount a volume, the security profile for the volume is preserved.

In previous versions of the operating system, protection for ODS-2 disk volumes was limited to the owner UIC and a protection code.

3.35.15 VMScluster Systems Require a Single Security Domain

V6.1

The vision for VMScluster systems is to provide a uniform computing environment which is highly scalable, highly available, and secure. Key to the success of this concept is the notion that there exists a single set of authorized users and that these users may have processes executing on any VMScluster member. Since its inception, OpenVMS has continually introduced new features in support of this concept.

When Digital introduced VMScluster systems, a primary component of the software technology was the distributed lock management services ($ENQ /$DEQ). These services provide a necessary framework for building distributed applications. The same version of the operating system included distributed file and record management subsystems along with a distributed Batch/Print subsystem, each coordinated through the use of the distributed lock manager.

Later versions of the operating system extended the process control system services to work in a clusterwide environment and introduced a new clusterwide security auditing subsystem. An improved Batch/Print subsystem represents a state-of-the art distributed application that runs simultaneously and transparently across all cluster members. In OpenVMS AXP Version Version T6.1-FT4, the security subsystem has been enhanced to ensure that all cluster-visible objects maintain consistent security profiles. Also, the system security auditing controls have been extended to work clusterwide.

The design center for all these features and for future OpenVMS operating system functionality has been and will remain that of a single security domain, with access control mediated by individual nodes using a common set of authorization information. This direction is necessitated by the design of the core cluster technology, which allows for the construction of highly available distributed applications. Substantial system management benefits also support this design.

In the single security domain model, a process, acting on behalf of an authorized individual, requests access to a cluster-visible object and a coordinating node determines the outcome by comparing its copy of the common authorization database with the security profile for the object being accessed. This model enforces security only when the authorization information and the object security profiles are consistent across all nodes in the cluster. Put another way, a single security domain is one in which each cluster member must make the same access control decision when presented with a particular user's access request for a particular object. OpenVMS AXP Version 1.5 provides this level of protection for files and queues, and Version Version T6.1-FT4 further incorporates all other cluster-visible objects: devices, volumes, and lock resource domains.

The OpenVMS operating system cannot enforce a level of separation needed to support different security domains on separate cluster members. Therefore, the OpenVMS VAX and OpenVMS AXP operating systems **do not** support multiple security domains.

The easiest way to ensure a single security domain is to maintain a single copy of each of the following files on one or more cluster-mounted disks. When a cluster is configured with multiple system disks, system logical names can be used to ensure that only a single copy of each file exists. The OpenVMS security domain comprises the following files:

SYS$MANAGER:AUDIT_SERVER.DAT (AXP V1.5 and VAX pre-V6.0 only)

SYS$MANAGER:VMS$AUDIT_SERVER.DAT (AXP V6.1 and VAX V6.0 and later)

SYS$SYSTEM:NETOBJECT.DAT

SYS$SYSTEM:NETPROXY.DAT

SYS$SYSTEM:QMAN$MASTER.DAT

SYS$SYSTEM:RIGHTSLIST.DAT[1]

[1] This file is optional, and does not exist unless it has been created by the system manager. There should be at most one copy of the file.

SYS$SYSTEM:SYSALF.DAT[1]

SYS$SYSTEM:SYSUAF.DAT

SYS$SYSTEM:SYSUAFALT.DAT[1]

SYS$SYSTEM:VMS$OBJECTS.DAT

SYS$SYSTEM:VMS$PASSWORD_HISTORY.DATA

SYS$SYSTEM:VMSMAIL_PROFILE.DATA

SYS$LIBRARY:VMS$PASSWORD_DICTIONARY.DATA

SYS$LIBRARY:VMS$PASSWORD_POLICY.EXE[1]

[1] This file is optional, and does not exist unless it has been created by the system manager. There should be at most one copy of the file.

Using shared files is not the only way of achieving a single security domain. Indeed, Digital fully recognizes the needs of its many existing customers who, for various reasons, already choose to use multiple copies of one or more of these files on different nodes in a cluster. For example, OpenVMS AXP customers may choose to deploy system-specific user authorization files (SYSUAF) to allow for different memory management working set quotas amongst different nodes. Such configurations are fully supported as long as the security information available to each node in the cluster is exactly the same.

The following sections describe the security-relevant portions of these files that must be synchronized across all cluster members to ensure that a single security domain exists. In the following list, files marked *required* contain some data that must be kept synchronized. Files marked *recommended* contain data that should be synchronized at the discretion of the site-security administrator or system manager. Nonetheless, Digital does recommend that recommended files be fully synchronized. Be aware that some of these files are created only on request and may not exist in all configurations. A file may be absent on one node only if it is absent on all nodes. As soon as any required file is created on one node, it must be created or commonly referenced on all remaining cluster members.

VMS$AUDIT_SERVER.DAT (and AUDIT_SERVER.DAT) [recommended]

This file contains information related to security auditing. Among the information contained in this file is the list of enabled security auditing events and the destination of the system security audit log file. When more than one version of this file exists, all copies should be updated after any SET AUDIT command. Failure to properly synchronize multiple versions of this file may result in partitioned auditing domains.

NETOBJECT.DAT [required]

This file contains the DECnet object database. Among the information contained in this file is the list of known DECnet server accounts and passwords. When more than one version of this file exists, all copies must be updated after any NCP {SET | DEFINE} OBJECT command. Failure to properly synchronize multiple versions of this file may result in unexplained network login failures and unauthorized network access.

NETPROXY.DAT [required]

This file contains the network proxy database. It is maintained by the OpenVMS Authorize utility. When more than one version of this file exists, all copies must be updated after any UAF proxy command. Failure to properly synchronize multiple versions of this file may result in unexplained network login failures and unauthorized network access.

QMAN$MASTER.DAT [required]

This file contains the master queue manager database. This file contains the security information for all shared Batch/Print queues. If two or more nodes intend to participate in a shared queuing system, a single copy of this file must be maintained on a shared disk.

RIGHTSLIST.DAT [required]

This file contains the rights identifier database. It is maintained by the OpenVMS Authorize utility and by various rights identifier system services. When more than one version of this file exists, all copies must be updated after any change to any identifier or holder records. Failure to properly synchronize multiple versions of this file may result in unauthorized system access and unauthorized access to protected objects.

SYSALF.DAT [required]

This file contains the system autologin facility database. It is maintained by the OpenVMS System Management utility. When more than one version of this file exists, all copies must be updated after any SYSMAN ALF command. Failure to properly synchronize multiple versions of this file may result in unexplained login failures and unauthorized system access.

SYSUAF.DAT [required]

This file contains the system user authorization file. It is maintained by the OpenVMS Authorize utility and is modifiable via the $SETUAI system service. When more than one version of this file exists, appropriate care must be taken to ensure that the fields in Table 3–3 are synchronized for each user record:

Table 3–3 Fields in SYSUAF and Associated $SETUAI Item Codes

Internal Name	$SETUAI Item Code
UAF$R_DEF_CLASS	UAI$_DEF_CLASS
UAF$Q_DEF_PRIV	UAI$_DEF_PRIV
UAF$B_DIALUP_ACCESS_P	UAI$_DIALUP_ACCESS_P
UAF$B_DIALUP_ACCESS_S	UAI$_DIALUP_ACCESS_S
UAF$B_ENCRYPT	UAI$_ENCRYPT
UAF$B_ENCRYPT2	UAI$_ENCRYPT2
UAF$Q_EXPIRATION	UAI$_EXPIRATION
UAF$L_FLAGS	UAI$_FLAGS
UAF$B_LOCAL_ACCESS_P	UAI$_LOCAL_ACCESS_P
UAF$B_LOCAL_ACCESS_S	UAI$_LOCAL_ACCESS_S
UAF$B_NETWORK_ACCESS_P	UAI$_NETWORK_ACCESS_P
UAF$B_NETWORK_ACCESS_S	UAI$_NETWORK_ACCESS_S
UAF$B_PRIME_DAYS	UAI$_PRIMEDAYS
UAF$Q_PRIV	UAI$_PRIV
UAF$Q_PWD	UAI$_PWD
UAF$Q_PWD2	UAI$_PWD2

(continued on next page)

Table 3–3 (Cont.) Fields in SYSUAF and Associated $SETUAI Item Codes

Internal Name	$SETUAI Item Code
UAF$Q_PWD_DATE	UAI$_PWD_DATE
UAF$Q_PWD2_DATE	UAI$_PWD2_DATE
UAF$B_PWD_LENGTH	UAI$_PWD_LENGTH
UAF$Q_PWD_LIFETIME	UAI$_PWD_LIFETIME
UAF$B_REMOTE_ACCESS_P	UAI$_REMOTE_ACCESS_P
UAF$B_REMOTE_ACCESS_S	UAI$_REMOTE_ACCESS_S
UAF$R_MAX_CLASS	UAI$_MAX_CLASS
UAF$R_MIN_CLASS	UAI$_MIN_CLASS
UAF$W_SALT	UAI$_SALT
UAF$L_UIC	<not applicable>

Failure to properly synchronize multiple versions of this file may result in unexplained login failures and unauthorized system access.

SYSUAFALT.DAT [required]

This file contains the system alternate user authorization file. This file serves as a backup to SYSUAF.DAT and is enabled via the SYSUAFALT system parameter. When more than one version of this file exists, all copies must be updated after any change to any authorization records in this file. Failure to properly synchronize multiple versions of this file may result in unexplained login failures and unauthorized system access.

VMS$OBJECTS.DAT [required]

This file contains the cluster-visible object database. Among the information contained in this file are the security profiles for all cluster-visible objects. When more than one version of this file exists, all copies must be updated after any change to the security profile of a cluster-visible object or after new cluster-visible objects are created. Cluster-visible objects include: disks, tapes, and resource domains. Failure to properly synchronize multiple versions of this file may result in unauthorized access to protected objects.

VMS$PASSWORD_HISTORY.DATA [recommended]

This file contains the system password history database. It is maintained by the system password change facility. When more than one version of this file exists, all copies should be updated after any password change. Failure to properly synchronize multiple versions of this file may result in a violation of the system password policy.

VMSMAIL_PROFILE.DATA [recommended]

This file contains the system mail database. This file is maintained by the OpenVMS Mail utility and contains mail profiles for all system users. Among the information contained in this file is the list of all mail forwarding addresses in use on the system. When more than one version of this file exists, all copies should be updated after any changes to mail forwarding. Failure to properly synchronize multiple versions of this file may result in authorized disclosure of information.

VMS$PASSWORD_DICTIONARY.DATA [recommended]

This file contains the system password dictionary. The system password dictionary is a list of English words and phrases that are not legal for use as account passwords. When more than one version of this file exists, all copies should be updated after any site-specific additions. Failure to properly synchronize multiple versions of this file may result in a violation of the system password policy.

VMS$PASSWORD_POLICY.EXE [recommended]

This file contains any site-specific password filters. It is created and installed by the site-security administrator or system manager. When more than one version of this file exists, all copies should be identical. Failure to properly synchronize multiple versions of this file may result in a violation of the system password policy.

3.35.16 Volumes and Devices Maintain Separate Access Control Lists

V6.1

In previous versions of the operating system, the security characteristics for devices and volumes were combined. When you mounted a volume on a device, the system combined the owner UIC and protection code for the volume with the ACL (if any) for the device. If you needed an ACL to control access to a volume, you had to set the ACL on the device first and then mount the volume. There was no separate ACL for the volume itself.

You can now define an ACL as part of a volume profile. It is the volume profile, and not the device profile, that controls access to a mounted volume. The device profile controls who can allocate or mount a volume on the device.

3.35.17 Terminal Ownership and Protection—Template Security Profile

V6.1

In previous versions of the operating system, the system parameter TTY_PROT set the default protection for all terminals in relation to the UIC specified by the system parameter TTY_OWNUIC.

In OpenVMS AXP Version Version T6.1-FT4, a template security profile sets the owner and protection of terminals. To display the default terminal template for your site, enter the following command:

```
$ SHOW SECURITY/CLASS=SECURITY_CLASS DEVICE.TERMINAL
```

To modify the site template, use the following command

```
$ SET SECURITY/CLASS=SECURITY_CLASS -
_$ /PROFILE=TEMPLATE=TERMINAL/PROTECTION=(PROTECTION_CODE) -
_$ /OWNER=NEW_UIC DEVICE
```

You cannot set security characteristics by using the DPT_STORE entries in device drivers. Any existing drivers with these entries do work correctly, however, security characteristics are taken from template profiles.

When drivers are only available in object form, the symbols TTY$GL_OWNUIC and TTY$GW_PROT can be defined in a link options file. For example,

```
    SYMBOL=TTY$GL_OWNUIC,80008214
    SYMBOL=TTY$GW_PROT,80008210
```

The actual value of each symbol must be an address that is readable in kernel mode, although the contents are irrelevent.

3.35.18 Security Profile Changes Distributed Clusterwide

V6.1

A VMScluster system is now a single security domain. If you change the security profile for a volume or a device mounted clusterwide, your changes are distributed to the other nodes in the cluster.

In previous versions of the operating system, changes to volume and device security profiles were not distributed to the other nodes in a cluster. You could use the SET VOLUME command to change the owner UIC or protection on a volume. SET VOLUME modified the security profile on disk and the security profile in memory. However, if the volume was mounted on another node in the cluster, the profile on that other node remained unchanged. To make a clusterwide change in the security profile, you had to dismount and remount the volume on the other nodes or enter the SET VOLUME command on the other nodes. This is no longer necessary.

3.35.19 SET ACL Qualifiers /LIKE and /DEFAULT—Change in Propagation of Protected and Hidden ACEs

V6.1

OpenVMS AXP Version Version T6.1-FT4 changes the way the DCL commands SET ACL/LIKE (superseded by the new command SET SECURITY/LIKE) and SET ACL/DEFAULT (superseded by SET SECURITY/DEFAULT) copy ACEs with the Protected and Hidden attributes. The changes are as follows:

- In previous versions of the operating system, the /LIKE qualifier and the /DEFAULT qualifier deleted the existing ACL (including Protected ACEs) on the target object before copying the ACL. In Version Version T6.1-FT4, ACEs with the Protected attribute are not deleted. In other words, you can copy an ACL to an object without overwriting any protected ACEs on the target object.
- In previous versions, the /LIKE qualifier did not copy ACEs with the Hidden attribute to the target object. In Version Version T6.1-FT4, the Hidden attribute does not prevent the ACE from being copied.

As a result of these changes, you can propagate an ACL without losing information that is protected or hidden. This is particularly useful for preserving ACEs that identify data-dependent information. For example, DDIF files have an ACE that identifies the file as DDIF. Utilities, such as TYPE, check the ACE before extracting text in the DDIF file. In previous versions of the operating system, these data-dependent attributes were lost (even if they were marked as protected or hidden) when you changed the ACL on a set of files. Without the ACEs, which provided data-dependent information, the files became inaccessible.

3.35.20 SET AUDIT/FAILURE_MODE=WAIT Is Obsolete

V6.1

The SET AUDIT command qualifier /FAILURE_MODE is now obsolete. The system no longer fails in its attempts to write a security alarm to a security terminal.

3.35.21 SET RIGHTS_LIST/ENABLE DCL—Default Set of Identifier Attributes

V6.1

Starting with OpenVMS AXP Version Version T6.1-FT4, the DCL command SET RIGHTS_LIST/ENABLE includes a default set of identifier attributes for the identifier being enabled or modified. The default attributes are taken from the associated holder record of the UIC identifier of the process issuing the command. If the identifier is not held by the process, the attributes default to the natural attributes (that is, the set of attributes associated with the identifier when it was first created). The SET RIGHTS_LIST command uses the /ATTRIBUTE qualifier to add or subtract attributes as necessary.

Prior to OpenVMS AXP Version Version T6.1-FT4, there was no default. All the required attributes had to be separately listed using the /ATTRIBUTES qualifier. System managers may want to search existing DCL command procedures for any SET RIGHTS_LIST/ENABLE commands and consider whether such commands still produce the desired effect. In most cases, simply removing any existing /ATTRIBUTES qualifiers will result in the desired behavior.

3.35.22 SET SECURITY and SHOW SECURITY Replace SET ACL and SHOW ACL

V6.1 In Version Version T6.1-FT4, the DCL commands SET SECURITY and SHOW SECURITY allow you to set and display security attributes with a single, security management interface. These commands replace several DCL commands, including SET ACL and SHOW ACL. When you use the new commands, note that they differ from the SET ACL and SHOW ACL commands in the following ways:

- SET ACL and SHOW ACL operate only on the ACL of an object. SET SECURITY and SHOW SECURITY operate on the whole security profile, including the owner and the UIC-based protection code. For example, SET ACL/DEFAULT or SET ACL/LIKE modify only the ACL for an object. SET SECURITY/DEFAULT and SET SECURITY/LIKE change the entire security profile.
- In keywords and qualifiers that identify the class of an object, SET SECURITY and SHOW SECURITY use the word CLASS. SET ACL and SHOW ACL use OBJECT_TYPE. For example:

```
$ SET SECURITY/CLASS=DEVICE DUA0: /ACL=(ID=JOE,ACCESS=NONE)
$ SET ACL/OBJECT_TYPE=DEVICE DUA0: /ACL=(ID=JOE,ACCESS=NONE)
```

- To create a new ACL that replaces an existing ACL, SET ACL uses the /NEW qualifier and SET SECURITY uses /DELETE=ALL. For example:

```
$ SET ACL SAFE.DAT/ACL=(ID=JOE,ACCESS=NONE)/NEW
$ SET SECURITY SAFE.DAT/ACL=(ID=JOE,ACCESS=NONE)/DELETE=ALL
```

- Both SET SECURITY and SET ACL provide a /LIKE qualifier, which specifies that the ACL of the target object is similar to the source. Both commands allow you to specify the name and class of the source object. However, the default class is different. If you do not specify the class of the source object, SET SECURITY uses the class of the target object. SET ACL uses FILE as the default class. For example:

```
$ SET SECURITY/CLASS=DEVICE DUA0: /LIKE=(CLASS=FILE,NAME=SAFE.DAT)
$ SET SECURITY/CLASS=DEVICE DUA1: /LIKE=NAME=DUA0:

$ SET ACL/OBJECT_TYPE=DEVICE DUA0: /LIKE=NAME=SAFE.DAT
$ SET ACL/OBJECT_TYPE=DEVICE DUA1: /LIKE=(OBJECT_TYPE=DEVICE,NAME=DUA0:)
```

3.36 Symmetric Multiprocessing

The following notes pertain to symmetric multiprocessing.

3.36.1 Symmetric Multiprocessing Extension License

V1.5 With Version 1.5 of the OpenVMS AXP operating system, you must register the Symmetric Multiprocessing (SMP) Extension License if you have an SMP system. This license upgrades the Operating System Base License and all Interactive User licenses (including Unlimited) to the matching multiprocessing level of your system.

Because the Symmetric Multiprocessing (SMP) Extension License grants all the rights the existing Base and User licenses provided at the uniprocessing level, you do not need to reinstall those licenses when you upgrade to a multiprocessing system. Each time you upgrade your system to a new multiprocessing level (for example, from a DEC 7000 Model 620 AXP system to a DEC 7000 Model 630 AXP system), you simply add an SMP Extension License to your existing licenses.

3.36.2 Restrictions

V1.5 With the following exceptions, the level of SMP functionality is identical to that found on VMS Version 5.4:

- Primary switching is not supported.
- The system parameter MULTIPROCESSING, which can be modified by both the System Generation utility (SYSGEN) and SYSBOOT, supports a new optional value of 4. This optional value guarantees that the streamlined system synchronization image is loaded, regardless of the multiprocessing capability or number of CPUs within the system.

  For additional information about the MULTIPROCESSING parameter's new optional value, see Section 3.5.4.3.

The Alpha AXP architecture adds additional constraints on the ordering and availability of memory accesses between CPUs in SMP configurations, requiring stricter synchronization requirements than those that work on VAX systems.

Ordering of memory references and atomicity problems cause unpredictable results if standard VAX SMP synchronization techniques are not used. However, any code sequences that use these techniques are adequately protected in the AXP environment without further modification. For additional information about synchronization techniques, see *Migrating to an OpenVMS AXP System: Planning for Migration.*

3.37 System Disks Connected to HSC Devices

V1.5 Digital recommends that you use high-priority HSC requestors when connecting system disks to HSC devices. Refer to the *HSC User Guide* for information about requestor priorities.

3.38 System Management Utility

V1.5 The following sections describe changes, known problems, and restrictions that apply to the System Management utility (SYSMAN).

3.38.1 SYSMAN to be Phased Out

V6.1

SYSMAN has been the general purpose system management utility for OpenVMS since VAX/VMS Version 5.0. It is useful as a single user interface providing numerous system management functions and distributed system management access, especially within VMSclusters.

Distributed system management technologies now being developed will eventually replace SYSMAN. As these technologies are included in OpenVMS systems, Digital intends to phase out SYSMAN. Note however, that none of SYSMAN will be removed until equivalent or better system management functionality is available to OpenVMS users.

It is likely that some SYSMAN components (for example, the DO command, PARAMETERS, and so forth) will be incorporated into new tools and utilities over several releases of OpenVMS. At the same time, other system management components that are not presently available through SYSMAN will be included in the new sets of tools and utilities.

OpenVMS will continue to provide all of the current SYSMAN functionality and the related character-cell and command procedure interfaces to SYSMAN components, but future user interfaces may not incorporate the current SYSMAN command langauge.

3.38.2 /CLUSTER Qualifier—Problem

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 AXP 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 AXP operating system.

3.38.3 No Secondary Error Messages for PARAMETERS SET and PARAMETERS SHOW Commands — Problem

V1.5

When you enter the PARAMETERS SHOW or PARAMETERS SET command and specify a parameter that does not exist on the AXP platform, SYSMAN does not return a secondary error message. The following example shows the informational primary message that SYSMAN currently returns when invalid parameters are specified:

```
SYSMAN> PARAMETERS SHOW KFILSTCNT
%SYSMAN-I-NODERR, error returned from node FLAM21
```

```
SYSMAN> PARAMETERS SET KFILSTCNT 16
%SYSMAN-I-NODERR, error returned from node FLAM21
```

The following example shows the secondary error message that SYSMAN should return:

```
SYSMAN> PARAMETERS SHOW KFILSTCNT
%SYSMAN-I-NODERR, error returned from node FLAM21
-SMI-E-NOSUCHPARM, no such parameter
```

```
SYSMAN> PARAMETERS SET KFILSTCNT 16
%SYSMAN-I-NODERR, error returned from node FLAM21
-SMI-E-NOSUCHPARM, no such parameter
```

Digital expects to fix this problem in a future release.

3.38.4 Performing Clusterwide DISKQUOTA Commands — Restriction

V1.5 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>
```

3.38.5 Ownership of System Directories

V1.5 During installation of OpenVMS AXP systems, the following user accounts are created on the system disk:

Account	Account UIC	Directory
FIELD	[1,10]	SYS$SYSROOT:[SYSMAINT]
SYSTEM	[1,4]	SYS$SYSROOT:[SYSMGR]
SYSTEST	[1,7]	SYS$SYSROOT:[SYSTEST]
SYSTEST_CLIG	[1,7]	SYS$SYSROOT:[SYSTEST]

All directories and all files for these accounts are owned by UIC [1,4]. However, the files remain accessible because the FIELD, SYSTEST, and SYSTEST_CLIG accounts have system UICs ([1,*]).

3.39 System Parameters

The following sections pertain to system parameters.

3.39.1 System Tuning—New System Parameter SYSTEM_CHECK

V6.1 A new system parameter, SYSTEM_CHECK, has been added to allow a single parameter to enable consistency checks on system operation.

The following table summarizes the default settings for related system parameters. Customized configurations of system checking can be selected by modifying these parameters.

	Earlier Field-Test Releases	Epsilon FT3	Final Customer Version
BUGCHECKFATAL	1	0	0
POOLCHECK	%X616400FF	0	0
MULITPROCESSING	2	3	3
SYSTEM_CHECK	n/a	1	0

Note

For OpenVMS AXP Version T6.1-FT4 field test, the SYSTEM_CHECK parameter is turned on, but for the final customer version, it will be off by default. As a result, field-test performance may be affected.

Refer to *OpenVMS AXP Version 6.1 New Features Manual* for more information on the SYSTEM_CHECK parameter.

3.40 Terminal Fallback Facility (TFF)

V1.0 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).

- 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 AXP TFF differs from OpenVMS VAX TFF in the following ways:

- On AXP systems, the TFF fallback driver is named SYS$FBDRIVER.EXE. On VAX systems, the TFF fallback driver is named FBDRIVER.EXE.
- *V6.1* On AXP systems, TFF is capable of handling 16-bit character fallback. The OpenVMS AXP fallback table library (TFF$MASTER.DAT) contains four more 16-bit character tables than the VAX library. Table 3–4 describes these additional tables.

Table 3–4 TFF Character Fallback Tables

Table Name	Base	Description
BIG5_HANYU	BIG5	BIG5 for CNS 11643 (SICGCC) terminal/printer
HANYU_BIG5	CNS	CNS 11643 (SICGCC) for BIG5 terminal/printer

(continued on next page)

Table 3–4 (Cont.) TFF Character Fallback Tables

Table Name	Base	Description
HANYU_TELEX	CNS	CNS 11643 for MITAC TELEX-CODE terminal
HANGUL_DS	KS	KS for DOOSAN 200 terminal

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

- On AXP 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.

Please refer to the *OpenVMS Terminal Fallback Utility Manual* for more information about the Terminal Fallback facility.

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.

3.40.1 TMSCP Tape Serving Support for AXP Systems

V6.1

With OpenVMS AXP Version T6.1-FT4, TMSCP tape serving is supported on AXP processors. AXP systems can serve TMSCP tapes to either AXP or VAX systems in a VMScluster. Prior to this release, only VAX processors could serve TMSCP tapes.

See *VMScluster Systems for OpenVMS* for more information about TMSCP tape serving.

3.41 UETP Notes

The following sections pertain to UETP (user environment test package).

3.41.1 UETP Documentation Relocated

V6.1

In previous OpenVMS releases, the documentation for UETP (the user environment test package) software was provided in the OpenVMS upgrade and installation manuals. With this OpenVMS release, UETP documentation has been moved to the *OpenVMS System Manager's Manual: Tuning, Monitoring, and Complex Systems.*

3.41.2 FDDI Devices Not Supported

V1.5

The UETP test package does not support the following FDDI devices:

DEFZA
DEFTA
DEFAA

3.41.3 Device Phase Access Violation

V6.1

The UETP device phase can have an access violation (ACCVIO) when running on a standalone system. As a temporary work-around and to avoid running into the problem, the system parameter SCSNODE must be set to the system nodename.

```
$ mcr sysgen
SYSGEN> set scsnode "your_nodename"
SYSGEN> write current
SYSGEN> exit
$  !make sure the system is rebooted.
```

3.41.4 Adding Special Accounts

V6.1

Effective with this release, the username SYSTEM is the only one which is automatically created as part of a fresh installation. In order to run the User Environment Test Program (UETP) usernames SYSTEST and SYSTEST_CLIG must be created. Some sites may also want to create a Field Service account for use by regularly assigned Field Service personnel.

To create accounts for UETP or Field Service, a new command procedure is provided which can be invoked with the command:

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

In contrast to former practice, this command procedure allows choice of the username for Field Service personnel. This allows assignment of separate usernames to separate individuals for better accountability and also reduces the risk of successful outside attacks against a known username such as FIELD.

Digital still recommends that privileged accounts described above be disabled when not in use and that generated passwords be used for Field Service accounts.

3.42 VMScluster Systems

The following sections contain information pertaining to VMScluster systems.

3.42.1 Intense File System Activity Related to LEF Hangs

V1.5

VMScluster systems may hang when all the following conditions are met:

- File system activity is extremely high (open, close, extend, truncate, and delete functions).
- All of the file system activity occurs on one volume, and the volume is full.
- All of this activity occurs across several VMScluster nodes.
- At least one VMScluster node has small XQP buffer caches.

This problem is frequently seen when heavily loaded UETP testing is conducted on recently configured systems.

To work around this problem, increase the setting of the system parameters listed in Table 3–5. The variable *n* represents the maximum number of processes expected in a system.

Table 3–5 System Parameters

System Parameter	Value
ACP_MAPCAHCE	Greater than *n*
ACP_DIRCACHE	Greater than (*n*x2)
ACP_HDRCACHE	Greater than (*n*x3)
ACP_DINDXCACHE	Greater than *n*

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

3.42.2 Recommended Versions of OpenVMS

V6.1 OpenVMS AXP Version T6.1-FT4 and OpenVMS VAX Version T6.1-FT4 provide two levels of support for mixed-version and mixed-architecture VMSclusters. These two support types are WARRANTED and MIGRATION.

Warranted support is provided for several pairs of OpenVMS VAX and OpenVMS AXP versions. Warranted support means that Digital has fully qualified the two versions coexisting in a VMScluster, and will answer all problems identified by customers using these configurations.

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

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

Note

Digital does not support the use of more than two versions in a VMScluster at a time. In many cases more than two versions will successfully operate but Digital cannot commit to resolving problems experienced with such configurations.

Figure 3–2 shows which level of support is provided for all possible version pairings.

Figure 3–2 Digital Support for Mixed-Architecture VMSclusters

(continued on next page)

Figure 3–2 (Cont.) Digital Support for Mixed-Architecture VMSclusters

```
              +-----------+-----------+-----------+-----------+
              | AXP V6.1  | VAX V6.1  | VAX V6.0  | AXP V1.5  |
+-------------+-----------+-----------+-----------+-----------+
| VAX V5.5-2  | Migration | Migration | Migration | WARRANTED |
+-------------+-----------+-----------+-----------+-----------+
| AXP V1.5    | Migration | Migration | WARRANTED |
+-------------+-----------+-----------+-----------+
| VAX V6.0    | Migration | Migration |
+-------------+-----------+-----------+
| VAX V6.1    | WARRANTED |
+-------------+-----------+
```

3.43 VMSINSTAL Checks Process Quotas

V1.5 At the beginning of a layered product installation, VMSINSTAL examines a number of process quotas in the installation account. VMSINSTAL determines whether the quotas are adequate for the installation of the product and notifies you if they are not.

Process quota values vary with the type and version of the operating system in use, as shown in Table 3–6.

Table 3–6 Minimum VMSINSTAL Installation Process Quotas

Process Quota	Minimum Value for OpenVMS VAX Version 6.0 [1]	Minimum Value for OpenVMS AXP Version 1.5 [1]
ASTLM	40	24
BIOLM	40	18
BYTLM	32768	32768
DIOLM	40	18
ENQLM	200	200
FILLM	300	100

[1] In future releases of the OpenVMS AXP operating system, minimum values may increase. Similarly, minimum values shown for OpenVMS VAX Version 6.0 may change in future releases.

A layered product may require additional process quotas or higher process quota values than those shown in Table 3–6. For specific information, see the layered product's installation guide.

Note

Use the Authorize utility (AUTHORIZE) to change process quota values. Changes that you make do not take effect until you log out and log in again.

3.44 VOID Kits Included on OpenVMS AXP Version 1.5 CD

V1.5 The following sections describe two updates that must be applied to OpenVMS VAX Version V5.5-2 systems. Both update save sets are located in the [VAX_KIT] directory on your OpenVMS AXP Version 1.5 CD. You must reboot your VAX systems in order to apply these fixes. These update kits are required for AXP and VAX system compatibility in a dual-architecture VMScluster environment.

3.44.1 Memory Exhaustion Problem on Clustered Systems

V1.5 Memory exhaustion occurs on VMScluster systems when a SYSAP that exists on a VAX machine is not running on the AXP nodes. To address this problem, an update kit named VAXSYSL09_U2055 is located in the [VAX_KIT] directory on your OpenVMS AXP Version 1.5 CD. A new SYS$LOADABLE_IMAGES:SYS$SCS.EXE image is supplied with the kit.

Installation Instructions

Use the VMSINSTAL utility to install this update kit. Log in to the SYSTEM account, and enter the following command:

```
$ @SYS$UPDATE:VMSINSTAL VAXSYSL09_U2055 [save-set location]
```

The save-set location can be a tape drive or a disk directory that contains the kit save set.

3.44.2 PEDRIVER Update Kit

V1.5 The PEDRIVER update kit addresses the following problems:

- A problem that manifests itself as anything from slow performance to broken virtual circuits (which may result in VAXport crashes).
- The XQDRIVER (for the DELQA and the DEQTA devices) does not report the correct number of receive buffers to PEDRIVER. (XQDRIVER is required for proper operation of PEDRIVER.)

The update kit, named VAXLAVC02_U2055, is located in the [VAX_KIT] directory on your OpenVMS AXP Version 1.5 CD.

The following files were patched or replaced:

- [SYS$LDR]PEDRIVER.EXE (new image)
- [SYS$LDR]XQDRIVER.EXE (new image)
- [SYSEXE]SDA.EXE (new image)
- [SYSEXE]SCSDEF.STB (new file)
- [SYSEXE]PEDRIVER.STB (new file)
- [SYSLIB]LIB.MLB (replaces $PEMCOMPDEF and $PEMCLSTDEF macros)
- [SYSHLP.EXAMPLES]LAVC$FAILURE_ANALYSIS.MAR (new file)

Note

Various internal structures changed as a result of the fixes to PEDRIVER. The System Dump Analyzer utility (SDA) and the SCS and PEDRIVER symbol tables are included in this kit to allow proper SDA operation. This also necessitated a change in [SYSLIB]LIB.MLB.

Installation Instructions

Use the VMSINSTAL utility to install this update kit. Log in to the SYSTEM account, and enter the following command:

```
$ @SYS$UPDATE:VMSINSTAL VAXLAVC02_U2055 [save-set location]
```

3.45 Volume Shadowing—Problem Fixed

V6.1 The following problem has been fixed for OpenVMS AXP Version T6.1-FT4:

If your terminal had a device type of UNKNOWN, FT1-FT8, or LN01K, a LICENSE LIST command would list all the PAKs in your license database (LDB) except for the last PAK.

3.46 Volume Shadowing Restrictions

V6.1 This section contains restrictions for Volume Shadowing for OpenVMS AXP.

3.46.1 Phase I Shadowing Not Supported

V6.1 Phase I (controller-based) volume shadowing is not supported on Volume Shadowing for OpenVMS AXP; only phase II is supported.

If other VAX nodes in your VMScluster are currently using phase I and you want to access shadowed data from an AXP system, you must migrate to phase II. Phase II (host-based) volume shadowing is designed to replace phase I shadowing with significantly enhanced data availability. Phase II supports a much wider range of configurations, disk controllers, and disks.

See Appendix A of *Volume Shadowing for OpenVMS* for instructions on how to convert from phase I to phase II, and for a more detailed description of phase I and phase II.

3.46.2 Setting System Parameter SHADOWING

V6.1 Volume Shadowing for OpenVMS AXP requires that you run phase II of the shadowing software. This means that you must set your SHADOWING system parameter to 2 after you install volume shadowing.

Chapter 3 of *Volume Shadowing for OpenVMS* contains more information about other shadowing parameters and a sample MODPARAMS.DAT file.

3.46.3 RZ57 Microcode Problem

V6.1 A problem exists with including RZ57 disks in shadow sets; any but the most recent version of microcode (Version 600) for the controllers is incompatible with volume shadowing. In addition, OpenVMS does not support the RZ57 disk drive.

You can, however, include systems with RZ57 disks in a cluster that runs volume shadowing, and you can run volume shadowing on systems that have RZ57 disks.

3.46.4 KZMSA Adapter Problem on DEC 7000 AXP Systems

V6.1 Disks on DEC 7000 AXP systems that are connected to a VMScluster through a KZMSA device are not protected against media failure from defects. A media defect refers to an imperfection in the oxide coating on the surface of the disk platter. Over time, the oxide degrades, resulting in a loss of data. This type of failure is extremely rare, but it is one that shadowing is expected to protect against.

You *can* put these disks in shadow sets, but this means that disks on this adapter will be removed from the shadow set in the rare occurrence of a media defect.

This problem will be corrected in the future.

3.46.5 VMSCluster Hangs on DISMOUNT/CLUSTER Command

V6.1

A known problem in the shadowing layered product may very rarely cause a clusterwide hang as a result of the DISMOUNT command. This problem happens when the DISMOUNT/CLUSTER command that you issued to a shadow set hangs. The process that issued the command may wait for a reply from a remote node in the cluster. Unfortunately, the process on this remote node holds an exclusive lock, preventing other nodes from dismounting the shadow set.

To work around this problem, reboot the nodes in the cluster that serve the devices included in the shadow set. As an alternative to using DISMOUNT /CLUSTER, you can use DISMOUNT/SYSTEM from each node in the cluster to dismount the shadow set.

3.46.6 Disks on the KDM70 Device Not Supported

V6.1

Disks attached to the cluster using the KDM70 controller are not supported in shadow sets. Devices on this type of controller experience a high number of "invalid MSCP command" errors under load. This results in their removal from the shadow set.

This problem will be corrected in the future.

3.46.7 Assisted Copy Operation Resets Incorrectly After Minimerge

V6.1

Volume shadowing may incorrectly reset an assisted copy operation after a system failure and a subsequent minimerge operation. If a node with the shadow set mounted fails at the time an assisted copy operation is in progress, the copy may restart at 0% copied.

The desired behavior is for the copy operation to continue at the percentage copied when the crash occurred. For example, if the shadow set was 33% copied at the time of the crash, the copy should resume at 33% after the system crash and the minimerge completes.

This problem will be fixed in a future version of the operating system.

3.47 Watch Chip — Change in Time Range

V1.0

AXP systems maintain their system time during power failures and system downtime 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 AXP computer, after running another operating system such as OSF/1, is rebooted using the OpenVMS AXP operating system.

3.48 XCDAC Adapter Storage-Only Support Available

V1.5 The XCDAC adapter (the XMI to CI adapter with an NPORT driver interface) is available in Version 1.5 for storage purposes only. A supported configuration consists of a dedicated star coupler, a single DEC 7000 Model 600 AXP computer, and multiple HSC controllers.

The XCDAC hardware is identical to the CIXCD hardware, except for the addition of new microcode. The XCDAC adapter is supported on all AXP XMI machines (DEC 7000 Model 600 AXP computers).

3.49 X Terminal Support

V1.5 This release of OpenVMS AXP supports remote X sessions over LAT lines. In Version 1.0, X terminals could create only local DECterm processes and could connect to an AXP system only like a terminal server. In Version 1.5, X terminals can create LAT X sessions.

To set up your system for X terminal use, enter the following commands after your system boots:

```
$ DEFINE /SYSTEM DECW$INSTALL_XTERMINAL TRUE
$ @SYS$MANAGER:DECW$STARTUP "XTERMINAL"
```

If you want to have X terminal setup performed automatically when the system boots, place the following line in the SYS$MANAGER:SYLOGICALS.COM file:

```
$ DEFINE /SYSTEM DECW$INSTALL_XTERMINAL TRUE
```

The DECwindows startup procedure at the end of the system boot invokes the SYS$STARTUP:DECW$STARTXTERMINAL.COM procedure, which in turn installs appropriate images and starts up the X terminal font daemon (the DECW$FD process).

Once the system boots, you can make a LAT X connection from an X terminal. Startup of a LAT X session is actually performed from the X terminal itself. The X terminal font daemon is also included with this release. As a result, you can set an X terminal's font path to an AXP system (via LAT).

With this release, you can use LAT as a transport to an X terminal in order to display a DECwindows application. The following example shows how to run a DECwindows application and display it on an X terminal (assuming security on the terminal is set appropriately):

```
$ SET DISPLAY /CREATE /TRANSPORT=LAT /NODE=LAT_08002Bxxxxxx
$ RUN SYS$SYSTEM:DECW$PUZZLE
```

In the example, *xxxxxx* is specific to the user's X terminal. The example shows how to set the display back to the X terminal and then run a DECwindows application (in this case, PUZZLE). The application is displayed on the specified X terminal.

For more information about X terminal configuration, refer to the X terminal documentation that accompanied your terminal.

4 Programming Release Notes

This chapter contains information of particular interest to programmers, such as tools, utilities, system services, and the file system.

4.1 Release Notes Specific to OpenVMS AXP Version T6.1-FT4

V6.1 The release notes in this document are cumulative from OpenVMS AXP Version 1.0. In this chapter, the following sections contain programming release notes that pertain specifically to OpenVMS AXP Version T6.1-FT4:

- TBS

4.2 Calling Standard Change Between Native and Translated Images

V1.5 OpenVMS AXP Version 1.0 did not support calls between translated and native images where a structure greater than 32 bits was passed by immediate value in a register argument.

The only language that naturally supports *receiving* such a call is C (VAX C or DEC C for OpenVMS). The following sections should be of interest to those who use such calls between native and translated images to routines written in the C programming language. It may also interest those who customize the autojacketing interface between the translated and native environments of OpenVMS AXP, such as constructing their own procedure descriptors. (Such work is usually done with MACRO-64 Assembler for OpenVMS AXP Systems.)

4.2.1 Changes to the Procedure Signature Information Block (PSIG)

V1.5 The following changes have been made to the procedure signature information block (PSIG):

- A new 64-bit argument, PSIG$K_RA_Q, has been added to the PSIG$_V_REG_ARG_INFO field. This new argument has a value of 1.

  The argument is passed in an integer register in the native environment and in 2 longwords within the argument list in the translated environment. The new capability treats a 64-bit segment of data (a portion of a structure) as a single register argument for the native environment and as two consecutive longword entries in the argument list for the translated environment. Compilers specify this behavior by using the new PSIG$K_RA_Q register argument code in the procedure signature information block.

  The corresponding capability for arguments numbered greater than 6 in the native environment (memory arguments) is already present. For the sake of consistency, however, a new symbolic name for the corresponding memory argument code is provided, as described in the next item.

- A new symbolic name, MASE$K_MA_Q, replaces the memory argument signature bit known as MASE$K_MA_I64. MASE$K_MA_Q is a 64-bit argument with a value of 0.

  The name MASE$K_MA_Q is preferable to the former name (MASE$K_MA_I64) because the new name makes no statement about the argument's being "integer" when in fact the code is used to support transmission of structures and floating-point values across the interface between the native and translated environments.

- Formerly, the symbolic name PSIG$K_RA_I64 in the PSIG$_V_REG_ARG_INFO field had the following two meanings:
  - 64-bit argument passed in an integer register.

    Using the symbolic name PSIG$K_RA_I64 for register arguments is no longer recommended because it does not accurately describe the behavior triggered by that code. When an argument is present, that code now results in behavior identical to that of PSIG$K_RA_I32.
  - Argument is not present.

    A new symbolic name, PSIG$K_RA_NOARG, is defined for the second meaning for which register argument code PSIG$K_RA_I64 was used. PSIG$K_RA_NOARG means that no argument is present as determined from the argument count for the call.

4.2.2 Versions of Compiled Languages

V1.5

Languages that make use of the new procedure signature register argument code have a method for expressing the notion of *passing* structures whose size is greater than 32 bits by immediate value.

The following initial versions of OpenVMS AXP compilers do not support this capability:

- DEC Ada Version 3.0 for OpenVMS AXP Systems
- DEC C Version 1.2 for OpenVMS AXP Systems
- DEC FORTRAN Version 6.0 for OpenVMS AXP Systems

Later versions of these compilers will be able to generate the new procedure signature register argument code in appropriate circumstances. The new procedure signature register argument code produces the new behavior only when run with OpenVMS AXP Version 1.5 or greater.

Digital suggests that, if you need to pass structures larger than 32 bits between the translated and native environments compatibly on both OpenVMS AXP Version 1.0 and later versions, pass those structures by reference. This is not a problem for MACRO-32 because that language does not provide any method for passing a structure larger than 32 bits by immediate value. Other languages not listed above should not encounter any problems with this issue because the first OpenVMS AXP official release of the language included this change (if they support passing a structure larger than 32 bits by immediate value).

4.2.3 MACRO-64 Assembler Version 1.0 for OpenVMS AXP Systems

V1.5 To use Version 1.0 of the MACRO-64 Assembler for OpenVMS AXP Systems and generate the new procedure signature register argument code, add the following code before invoking any of the regular calling standard macros:

```
                                ; Change MA_I64 to use new MA_Q code (MA_Q = 1).
                                ; (Make MA_I64 pretend to be MA_Q).
MACRO64$CALLSTD_DEFS            ; Do initial V1.0 definitions.
MACRO64$RA_I64 = 1              ; Change RA_I64 to use new RA_Q code.
MACRO64$RA_i64 = 1              ; Lowercase version too.

                                ; Correct V1.0 call signature encodings.
MACRO64$REGARGTYPE_A    = "I32" ; VAX addresses are longword.
MACRO64$REGARGTYPE_a    = "I32" ; Lowercase too.
MACRO64$MEMARGTYPE_F    = "I32" ; F_float is a longword in length.
MACRO64$MEMARGTYPE_f    = "I32" ; Lowercase too.
MACRO64$MEMARGTYPE_S    = "I32" ; S_float is a longword in length.
MACRO64$MEMARGTYPE_s    = "I32" ; Lowercase too.
MACRO64$MEMARGTYPE_UL   = "I32" ; U32 is a longword.
MACRO64$MEMARGTYPE_ul   = "I32" ; Lowercase too.
MACRO64$MEMARGTYPE_A    = "I32" ; VAX addresses are longword.
MACRO64$MEMARGTYPE_a    = "I32" ; Lowercase too.
```

This code also corrects some other procedure signature block generation errors when you use the MACRO-64 Assembler Version 1.0 for OpenVMS AXP Systems to generate calls to translated VAX images. (That is, when you invoke the $CALL macro with the TIE=TRUE argument.)

These changes do *not* cause $ROUTINE and $PROCEDURE_DESCRIPTOR to accept ARGLIST=Q where it now accepts ARGLIST=I64. You must still use I64 with $ROUTINE and $PROCEDURE_DESCRIPTOR, even though it will become obsolete.

With any subsequent release of MACRO-64 Assembler for OpenVMS AXP Systems, Digital plans to recommend changing use of I64 with $ROUTINE and $PROCEDURE_DESCRIPTOR to Q (issuing an informational message for I64 to that effect). $CALL changes will not be necessary because $CALL maps from an argument qualifier to the appropriate argument type.

For more information about the procedure signature information block, see the *OpenVMS Calling Standard*.

4.3 DEC C RTL Known Problems

V1.5 The following problems exist in the DEC C RTL in OpenVMS AXP Version 1.5:

- The C RTL loses carriage-control information when it reads from mailboxes. As a result, information that is read from a mailbox has no line breaks or carriage returns; the original lines are concatenated.
- When an application sets the reentrancy mode to AST, the C RTL always enables asynchronous system traps (ASTs) during a file open operation. To make your application disable ASTs, place SYS$SETAST(0) calls after every file open call and after the first I/O operations to the stdin, stdout, and stderr functions (that is, when the C RTL performs a file open operation to initialize the stdio device). This problem does not occur in any other reentrancy mode.

4.4 Run-Time Support for DEC COBOL Programs — New Feature

V1.5 OpenVMS AXP Version 1.5 includes run-time system support for programs compiled with DEC COBOL Version 1.0. DEC COBOL Version 1.0 is a compiler for the COBOL language producing high-performance object code. Refer to the product description and documentation for DEC COBOL Version 1.0 for a full list of features supported by that product.

By including the run-time system support with OpenVMS AXP, user programs and shareable images compiled with DEC COBOL Version 1.0 can execute on systems that are not licensed for the DEC COBOL Version 1.0 compiler.

4.5 Condition Handling—SS$_PAGRDERR Condition

V6.1 The signal argument of the SS$_PAGRDERR condition has been changed. Prior to OpenVMS AXP Version T6.1-FT4, the first signal argument was the mask of the transition not valid reason. In OpenVMS AXP Version T6.1-FT4, this argument has been changed to provide the actual I/O failure status that caused the page read error.

Condition Name	Explanation	
SS$_PAGRDERR	Type:	Fault
	Description:	Read error occured during an attempt to read a faulted page from disk.
	Arguments:	1. I/O failure status.
		2. Virtual address of referenced page.

4.6 Debugging and Image Activation

V6.1 Due to enhancements to the OpenVMS Debugger (DEBUG), the image activator has been modified to automatically activate SYS$SHARE:SYS$SSISHR.EXE when an image is RUN/DEBUG or linked with 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.

4.7 OpenVMS Debugger

The following sections describe corrections, restrictions, and known problems that are present in the OpenVMS Debugger.

The Delta/XDelta debugger (DELTA/XDELTA) is also available for debugging. See the *OpenVMS Delta/XDelta Debugger Manual* for details on using the Delta/XDelta utility.

4.7.1 Corrections to Known Problems

V6.1 In Version 1.5, if you set a breakpoint in an AST routine, and an AST fired during the execution of a STEP command, the STEP command incorrectly finished when the breakpoint in the AST routine triggered. When the AST routine finished and control returned to the point where the AST fired, the debugger reported an access violation at a random address.

This problem has been corrected.

4.7.2 Known Problems and Restrictions

V6.1 This section describes known problems and restrictions that affect the debugger in OpenVMS AXP Version T6.1-FT4. For complete information on debugger functionality, see the *OpenVMS Debugger Manual*.

4.7.2.1 Debugging Critical Sections

V1.5 The debugger does not support attempts to debug critical sections that are delimited by memory-locking (LDx_L) and memory-unlocking (STx_C) instructions. These critical sections must execute without interruptions, or they will fail. The exception mechanisms used by the debugger can interrupt a critical section.

The debugger warns you and surrenders control when it detects that a critical section is about to be interrrupted. To allow the critical section to succeed, perform the following steps:

1. Deactivate all breakpoints that are set on locations bounded by (and including) the memory-locking and -unlocking instructions.
2. Deactivate all nonstatic watchpoints.
3. Set a temporary breakpoint on the instruction following the memory-unlocking instruction, and type GO to reach the breakpoint.
4. Restore the breakpoints and watchpoints you disabled in steps 1 and 2.
5. Type GO to resume execution.

See the *Alpha Architecture Reference Manual* for more information on memory-locking and -unlocking instructions.

4.7.2.2 CALL Command Problems and Restrictions

V1.5 The following CALL command problems and restrictions appear in this release:

- When you issue a CALL command before another outstanding CALL command has finished (that is, before the "value returned" message appears), the debugger hangs. To work around this restriction, continue execution of the program with the STEP or GO commands until the previous CALL command finishes.
- You cannot currently pass parameters in the CALL command.
- You cannot debug routines that are activated *before* the routine activated by a CALL command. For example, suppose your program is stopped in routine MAIN, and you set a breakpoint in routine SORT. You issue the debugger CALL SORT command. While debugging routine SORT, you cannot debug routine MAIN. You must first return from the call to routine SORT.

- 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. In a future release of the operating system, display of these frames will be suppressed, so that behavior on an AXP system parallels the behavior on a VAX system.

4.7.2.3 EVALUATE Command Limitation

V1.5 When you issue the EVALUATE command for an integer, the debugger truncates the return value if it is larger than a longword.

4.7.2.4 EXAMINE Command Problems and Restrictions

V1.5 The following EXAMINE command problems and restrictions appear in this release:

- The first longword of a routine cannot be examined.
- When program execution is suspended in system space, the command EXAMINE PC displays an incorrect value for the high-order longword of the PC.
- If you issue an EXAMINE/PS command, the debugger returns a syntax error. To work around this problem, reenter the command as EXAMINE PS.
- If you issue an EXAMINE/INSTRUCTION command for a routine, the debugger displays incorrect instructions. To work around this problem, enter the following commands:
  - SHOW SYMBOL/ADDRESS *routine-name* to determine the correct code address for the routine
  - EXAMINE/INSTRUCTION *code-address-value* to display the correct instructions
- When the symbol to which an offset is applied is a routine, module, or image name, the debugger does not compute symbol offsets correctly. For example, if CALCULATE is a routine name, then locations of the form CALCULATE + *n* have an incorrect value computed for *n*.
- If you issue the command EXAMINE LABEL[*n*] or EXAMINE LABEL(*n*), where LABEL is a label for a code location and *n* is an integer, an access violation error results.

4.7.2.5 SET WATCH Command and Nonstatic Watchpoints

V1.5 By default, the SET WATCH command sets a nonstatic watchpoint. If you issue the SET WATCH command, the debugger treats the value as nonstatic, regardless of where the watched value is stored.

4.7.2.6 SET WATCH/STATIC Command and Static Watchpoints

V1.5

To set static watchpoints, append the /STATIC qualifier to the SET WATCH command. On AXP systems, the debugger displays the following informational message when you issue a SET WATCH/STATIC command:

```
%DEBUG-I-ALPHANOSSI, static watchpoints may cause memory probes to fail
```

The debugger implements static watchpoints by write-protecting the page containing the variable to be watched. When static watchpoints are set, a system service or user-written routine that writes to user memory can fail as it probes for write access, instead of completing successfully as it should.

On VAX systems, this situation is avoided through system service interception. On AXP systems, system service interception has not yet been implemented.

4.7.2.7 SET BREAK/UNALIGNED_DATA Command and Related System Service Call

V1.5

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.

4.7.2.8 SET TASK/ACTIVE Command and Spurious Errors

V6.1

The OpenVMS Debugger command SET TASK/ACTIVE does not work.

Setting a task with the /ACTIVE qualifier is a complicated maneuver in which the OpenVMS Debugger requests a state-change from DECthreads and then releases control of the program to allow DECthreads to make that change. Once DECthreads switches to the proper thread context, it signals the OpenVMS Debugger to take control again.

Some recent changes in handling special-case thread context switching appear to have upset this functionality. Digital will consider how to address this problem in a future release. In the meantime, be aware that using the SET TASK/ACTIVE command will result in spurious errors that may terminate your debugging session.

Using the SET TASK/ACTIVE command is usually not required. For example, with query-type actions, such as examining local variables, you can use the SET TASK/VISIBLE command. For an effective way to get control to STEP in a particular thread, use breakpoints strategically along with the SET TASK/HOLD command (instead of relying on SET TASK/ACTIVE).

4.7.2.9 Kept Debugger Restrictions and Problems

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. At DCL level, check to see whether any debugger subprocesses exist by issuing the command SHOW PROCESS /SUBPROCESS. If any debugger subprocesses exist, you can stop them by issuing 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.

V6.1

- The RUN/COMMAND="command-line" capability does not use the command table of the process from which the debugger was invoked. When you append the /COMMAND qualifier to the RUN command, the following error displays:

  ```
  DBG>  RUN/COMMAND="MY_COMMAND/MY_QUAL KIT_BUILD/COPIES=0"
  %DCL-W-IVVERB, unrecognized command verb - check validity and spelling
      \MY_COMMAND\
  ```

  To workaround this problem, ask your system manager to build a custom DCLTABLES file for your project. Note that the DCLTABLES file must be updated if you change the DCL command.

- When you run a sequence of many large programs, the debugger might fail due to exhaustion 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 changes back to its former state once the program has finished.

- If you are using the Motif interface (as opposed to character-cell screen mode), and you try to run a program that doesn't 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
  ```

  This is because 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 via a SPAWN/NOWAIT command. Therefore, you must press the Return key twice on the DECterm window from which the debugger was run after the debugger version number has appeared in the Source Window.

  Optionally, you can execute the Kept Debugger in the following manner:

  ```
  $ DEFINE DBG$INPUT NL:
  $ SPAWN/NOWAIT RUN SYS$SHARE:DEBUGSHR.EXE
  ```

- If you issue the RERUN command while your file is locked by another user, the debugger returns the following message:

  ```
  %DEBUG-E-NORERUNPGM, There is no program to RERUN
  ```

  To work around this problem, wait until your file is unlocked. Then, issue a RUN command and reset breakpoints.

4.7.2.10 DECwindows Motif Interface Restrictions and Problems

V6.1

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 which has gotten control of the mouse pointer via a PointerGrab or a KeyboardGrab, your workstation will hang.

  To work around this problem, debug your program using two workstations. For more information, see the *OpenVMS Debugger Manual.*

- Occasionally, if you are scrolling or clicking on items in the Register View, the following message displays in the DECterm window from which you initiated the debugging session:

  ```
  X Toolkit Warning: Not all children have same parent
     in XUnmanageChildren
  ```

  You can ignore this message.

- Initially, the debugger main and optional views windows in T6.1 may appear to be oddly sized. The resource file shipped with the Version 1.5 debugger causes these windows to take the shape of the previous source and control windows in Version 1.5.

  To work around this problem, resize the new windows and save your window configuration.

- If you attempt to typecast or change the radix for a monitored item (for example, a task in a multitasking program) where deposit operations do not make sense, the debugger does not issue an error message. Instead, it tries to complete the operation and then freezes the display.

  To workaround this problem, exit the debugger by issuing a Ctrl-Y key sequence from the DECterm in which you invoked the debugger.

- 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
DISPLAY	SET MODE [NO]SCREEN

(continued on next page)

Table 4–1 (Cont.) Debugger Commands Disabled in the DECwindows Motif Interface

EXAMINE/SOURCE
EXPAND
EXTRACT
HELP
MOVE
SAVE
SCROLL
SEARCH
SET MODE [NO]SCREEN_LOG
SET MODE [NO]SCROLL
SET OUTPUT [NO]TERMINAL
(SET,SHOW) SEARCH
(SET,SHOW) TERMINAL
(SET,SHOW) WINDOW
(SHOW,CANCEL) DISPLAY
SHOW SELECT

- Motif does not provide for specialized key support (such as Ctrl/Y), but the Motif interface provides alternative means of executing these functions (for example, the STOP button). Therefore, commands that require specialized key support are disabled in the Motif interface.
- With the exception of the PREVIOUS and NEXT keys, commands related to the keypad or key bindings apply only to the character-cell command interface. This functionality is disabled in the Motif interface.
- Commands related to character-cell terminal display apply only to the command interface. These commands are disabled in the Motif interface.
- The Motif interface to the debugger does not make use of a DECterm. Therefore, commands that require a DECterm are disabled in the Motif interface. The exception is the EDIT command, which is still available.
- If you decrease the size of the Communications Pane so that the prompt is occluded, the message area is not automatically scrolled to display the prompt.

  To work around this problem, use the vertical scroll bar to scroll the view so that the prompt reappears.
- Under certain circumstances, when the Breakpoint View option is invoked prior to setting a breakpoint, the Breakpoint View has a height of zero.

  To work around this problem, resize the Breakpoint View after it has been displayed and then resave its location.
- The column labels within the Monitor View and Tasking View do not line up properly when resizing the Monitor View.

4.7.2.11 Debugging Translated Images

V1.0 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 the Delta/XDelta Debugger, see the *OpenVMS Delta/XDelta Debugger Manual.*

Using the OpenVMS Debugger in a process where any translated shareable image (for example, EDTSHR) is active is not supported.

V6.1 If you are running a translated image that was linked /DEBUG or run /DEBUG, the system service calls SYS$NATIVE_TO_TRANSLATED and SYS$TRANSLATED_TO_NATIVE, used by the TIE, will not function properly.

To alleviate this problem, define the logical name SSI$AUTO_ACITVATE to "OFF".

4.7.2.12 Debugging Installed Resident Images

V1.0 The debugger does not support attempts to debug installed resident images.

4.7.2.13 Debugging Inlined Routines

V1.0 The debugger does not support attempts to debug inlined routines. If you attempt to debug an inlined routine, the debugger issues a message stating 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.

4.7.2.14 Debugging Global Sections

V1.0 The debugger does not support setting watchpoints on variables whose addresses are in global sections. If you attempt to set a watchpoint on a location in a global section, the debugger issues a message rejecting the watchpoint, as shown in the following example:

```
%DEBUG-E-BADWATCH, cannot watch protect address 'address-value'
```

4.7.2.15 Debugging Register Frame Procedures or Null Frame Procedures

V1.0 The debugger does not fully support attempts to debug Register Frame procedures or Null Frame procedures. If you issue the STEP/OVER or STEP/RETURN commands for these procedures, unexpected results might occur. For more information on Register Frame procedures and Null Frame procedures, see the *OpenVMS Calling Standard.*

4.7.2.16 Complex Variables in DEC Fortran Programs

V1.0 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.

4.7.2.17 Concealed Rooted-Directory Logical Names for Source Files

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 are unable to use the debugger SET SOURCE command to specify the new location of the source file.

4.7.2.18 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"
```

4.7.2.19 SET IMAGE Command Limitation

V1.0 For some large programs with many program sections (usually caused by many DEC Fortran routines with many COMMON blocks), the debugger may receive an internal error while processing a SET IMAGE command. In such cases, the image cannot be debugged.

4.7.2.20 SHOW BREAK and SHOW TRACE Command Limitations

V1.0 SHOW BREAK and SHOW TRACE 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).

4.7.2.21 STEP/INTO Command and User Exception Handlers

V1.0 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.

4.7.2.22 STEP/OVER Command Error with Recursive Calls

V1.0 When you issue a STEP/OVER command at a recursive call to the routine that is currently executing, a SYSTEM-F-OPCDEC error, an internal debugger error, or an access violation results. To work around this problem, either issue the STEP/OVER command at the initial call to the recursive routine or issue the STEP/INTO command at each recursive call of the routine.

4.7.2.23 STEP/OVER Command Error with One-Line Program Loops

V1.0 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 square 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);
```

4.7.2.24 $WAKE Call Followed by $HIBER Call

V1.0 If a program running under the two-process or multiprocess debugger issues a $WAKE call followed by a $HIBER call, the user application hibernates.

4.7.2.25 DCL ANALYZE/PROCESS_DUMP Command and Zero Program Counter

V1.0 The debugger cannot analyze a process dump file if the dump describes an exception where the program counter (PC) is zero.

4.7.2.26 Changed Watchpoint Locations During System Service Calls

V1.0

If a watched location changes during a system service routine, you are notified that the watchpoint occurred. However, the stack shows 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 return to your program.

4.7.2.27 STEP Command into System Service Calls

V1.0

If you attempt to single-step into a system service call, an access violation results.

4.7.2.28 LCK$M_DEQALL Modifier in $DEQ System Service Call

V1.0

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
```

4.8 DECdtm Services—New Condition Values Returned

V6.1

New condition values can now be returned by the DECdtm system services.

4.8.1 $ABORT_TRANS—New Condition Values Returned

V6.1

The $ABORT_TRANS system service now also returns the following condition values, in addition to the values that it returned previously:

- SS$_BADREASON
- SS$_CURTIDCHANGE
- SS$_NOTORIGIN
- SS$_TPDISABLED

For a full description of the $ABORT_TRANS system service, see the *OpenVMS System Services Reference Manual.*

4.8.2 $END_TRANS—New Condition Values Returned

V6.1

The $END_TRANS system service now also returns the following condition values, in addition to the values that it returned previously:

- SS$_CURTIDCHANGE
- SS$_NOLOG
- SS$_NOTORIGIN
- SS$_TPDISABLED

For a full description of the $END_TRANS system service, see the *OpenVMS System Services Reference Manual.*

4.8.3 $START_TRANS—New Condition Values Returned

V6.1 The $START_TRANS system service now also returns the SS$_CURTIDCHANGE condition value, in addition to the values that it returned previously.

For a full description of the $START_TRANS system service, see the *OpenVMS System Services Reference Manual.*

4.9 DECmigrate

V1.5 DECmigrate Version 1.1 software generates translated images that are executable on OpenVMS AXP Version 1.5 or later. This version of DECmigrate does not support OpenVMS AXP Version 1.0. Images translated by DECmigrate Version 1.0 are executable on all versions of the OpenVMS AXP system.

4.10 DECthreads

The following sections pertain to DECthreads.

4.10.1 Exit Handler Routines

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 subsequent release of the OpenVMS operating system.

4.10.2 DEC C RTL Problem

V1.5 A potential problem occurs between the DECthreads software and the DEC C RTL when a C program that calls LIB$FIND_IMAGE_SYMBOL dynamically activates DECthreads or another shareable image that depends on DECthreads. This problem may result in an access violation inside CMA$RTL in a call from DECC$SHR.

To work around this problem, link the image that calls LIB$FIND_IMAGE_SYMBOL against CMA$RTL.

Dynamically activating DECthreads or shareable images that depend on DECthreads does not work under all circumstances. Several software facilities contain code whose behavior is conditional upon the presence or absence of threads at the time that these facilities are initialized. If any of these facilities are in use by a program, threads cannot be safely brought into the process after execution begins because these facilities cannot respond to the change. An attempt to bring threads in at this stage typically causes the program to abort. Other unpredictable behavior, such as access violations, may result instead of program abortion.

4.10.3 Problem

V1.0 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 AXP.

The *Guide to DECthreads* provides additional detailed information about using DECthreads software.

4.11 Device Support on OpenVMS AXP

The following sections contain notes about OpenVMS AXP device drivers.

4.11.1 New Device Driver Interface

V6.1 OpenVMS AXP Version T6.1-FT4 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 that was provided in OpenVMS AXP Versions 1.0 and 1.5. The Step 2 driver interface provides support for writing device drivers in the C programming language or another high-level language and that can conform to the OpenVMS AXP calling standard. An example device driver written in C is provided in the SYS$EXAMPLES directory.

The following AXP device support manuals are available for FT4:

- *Creating an OpenVMS AXP Step 2 Device Driver from a Step 1 Device Driver*
- *Creating an OpenVMS AXP Step 2 Device Driver from an OpenVMS VAX Device Driver*
- *OpenVMS AXP Device Support: Developer's Guide*
- *OpenVMS AXP Device Support: Reference*

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*. To find out how to convert an OpenVMS VAX device driver to a Step 2 driver, see *Creating an OpenVMS AXP Step 2 Device Driver from an OpenVMS VAX Device Driver*. For information about how to write an OpenVMS AXP device driver in a high-level language, see the *OpenVMS AXP Device Support: Developer's Guide*. *OpenVMS AXP Device Support: Reference* contains reference information that pertains to each of these books.

4.11.2 All Step 2 Drivers Must Be Recompiled and Relinked

V6.1 You must recompile and relink any Step 2 driver that was compiled and linked on FT3 to load it successfully on FT4. If you attempt to load a Step 2 driver that was compiled and linked on FT3, the Sysman Utility will issue the following warning message, and the driver will not be loaded:

```
$ mcr sysman io connect mya0/noadapter/driver=$users:[jones]mydriver_ft3
%SYSMAN-I-NODERR, error returned from node BEAMME
-SYSTEM-F-DRVNOTVALID, device driver failed DPT consistency checks
```

You must also recompile and relink any Step 2 driver that was compiled and linked on FT1 or FT2 to load it successfully on FT4. If you attempt to load a Step 2 driver that was compiled and linked on FT1 or FT2, the Sysman Utility will issue the following warning message, and the driver will not be loaded:

```
$ mcr sysman io connect mya0/noadapter/driver=$users:[jones]mydriver_ft2
%SYSMAN-I-NODERR, error returned from node BEAMME
-SYSTEM-W-SYSVERDIF, system version mismatch; please relink
```

4.11.3 Step 1 Drivers Are Obsolete

V6.1

Step 1 driver interfaces were removed at FT3, and Step 1 drivers are not loadable. 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 OpenVMS AXP Versions 1.5 or 1.0, FT1, or FT2, the Sysman Utility 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 AXP operating system, the Linker will issue warnings 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.

4.11.4 Obsolete Step 1 Driver Support Routines

V6.1

The Step 1 JSB-Based Routines listed in this section are now obsolete. For information about the Step 2 routines that replace them, see *Creating an OpenVMS AXP Step 2 Device Driver from a Step 1 Device Driver.*

ACP$ACCESS
ACP$ACCESSNET
ACP$DEACCESS
ACP$MODIFY
ACP$MOUNT
ACP$READBLK
ACP$WRITEBLK
COM$SETATTNAST
COM$SETCTRLAST
EXE$ABORTIO
EXE$FINISHIO
EXE$FINISHIOC
EXE$IORSNWAIT
EXE$KP_STARTIO
EXE$LCLDSKVALID
EXE$MODIFY
EXE$MODIFYLOCK
EXE$MODIFYLOCK_ERR
EXE$ONEPARM
EXE$QIOACPPKT

EXE$QIODRVPKT
EXE$QIORETURN
EXE$READ
EXE$READCHK
EXE$READCHKR
EXE$READLOCK
EXE$READLOCK_ERR
EXE$SENSEMODE
EXE$SETCHAR
EXE$SETMODE
EXE$WRITE
EXE$WRITECHK
EXE$WRITECHKR
EXE$WRITELOCK
EXE$WRITELOCK_ERR
EXE$ZEROPARM
IOC$CLONE_UCB
IOC$LINK_UCB
MT$CHECK_ACCESS

4.11.5 Obsolete Data Structure Cells

V6.1

Some DDT and DPT data structure fields that supported Step 1 device drivers were removed at FT3. Table 4–2 lists the obsolete Step 1 fields and the Step 2 fields that have similar functions.

Note that the Step 2 cells use different names because they point to routines whose interfaces are different or they point to data structures whose layout is significantly altered. For this reason, do not replace each reference to an obsolete Step 1 field with its corresponding Step 2 field without taking into account the routine interface and data structure changes.

Table 4–2 Obsolete Data Structure Cells

Obsolete Step 1 field	Similar Step 2 field
DDT$L_ALTSTART	DDT$PS_ALTSTART_2 or DDT$PS_ALTSTART_JSB
DDT$PS_ALTSTART	DDT$PS_ALTSTART_2 or DDT$PS_ALTSTART_JSB
DDT$L_CANCEL	DDT$PS_CANCEL_2
DDT$PS_CANCEL	DDT$PS_CANCEL_2
DDT$L_CANCEL_SELECTIVE	DDT$PS_CANCEL_SELECTIVE_2
DDT$PS_CANCEL_SELECTIVE	DDT$PS_CANCEL_SELECTIVE_2
DDT$L_CHANNEL_ASSIGN	DDT$PS_CHANNEL_ASSIGN_2
DDT$PS_CHANNEL_ASSIGN	DDT$PS_CHANNEL_ASSIGN_2
DDT$L_CLONEDUCB	DDT$PS_CLONEDUCB_2
DDT$PS_CLONEDUCB	DDT$PS_CLONEDUCB_2
DDT$L_CTRLINIT	DDT$PS_CTRLINIT_2
DDT$PS_CTRLINIT	DDT$PS_CTRLINIT_2

(continued on next page)

Table 4–2 (Cont.) Obsolete Data Structure Cells

Obsolete Step 1 field	Similar Step 2 field
DDT$L_FDT	DDT$PS_FDT_2
DDT$PS_FDT	DDT$PS_FDT_2
DDT$L_MNTVER	DDT$PS_MNTVER_2
DDT$PS_MNTVER	DDT$PS_MNTVER_2
DDT$L_REGDUMP	DDT$PS_REGDUMP_2
DDT$PS_REGDUMP	DDT$PS_REGDUMP_2
DDT$L_START	DDT$PS_START_2 or DDT$PS_START_JSB
DDT$PS_START	DDT$PS_START_2 or DDT$PS_START_JSB
DDT$L_UNITINIT	DDT$PS_UNITINIT_2
DDT$PS_UNITINIT	DDT$PS_UNITINIT_2
DPT$PS_DELIVER	DPT$PS_DELIVER_2

4.11.6 Device Timeout Routine Address

V6.1

OpenVMS AXP Version T6.1-FT4 includes a change to the handling of device interrupt timeouts. Prior to this release, the UCB$L_FPC cell in the device Unit Control Block (UCB) contained the procedure value of the routine that served as both the resume from interrupt routine and the interrupt timeout routine. These routines are now separate. The new UCB cell UCB$PS_TOUTROUT is used for the procedure value of the interrupt timeout routine.

These changes are transparent to code that uses the WFIKPCH or WFIRLCH macros, or calls the IOC$PRIMITIVE_WFIKPCH or IOC$PRIMITIVE_WFIRLCH routines.

However, code that manually sets the UCB$V_TIM bit in the UCB$L_STS will now need to place the timeout routine procedure value into the UCB$PS_TOUTROUT cell instead of the UCB$L_FPC cell. For more information, see *Creating an OpenVMS AXP Step 2 Device Driver from a Step 1 Device Driver* and *OpenVMS AXP Device Support: Reference.*

4.11.7 Resuming a Kernel Process from a Device Interrupt Service Routine

V6.1

A device interrupt service routine can use one of two methods to resume a kernel process thread that has been suspended by the IOC$KP_WFIKPCH or IOC$KP_WFIRLCH routines. The preferred method is to the call EXE$KP_RESTART routine. This method is unchanged in FT4.

The second method is to load R3 and R4 from the UCB fork block and then transfer control via a JSB to the routine whose procedure value is in UCB$L_FPC(R5).

```
MOVL    UCB$Q_FR3 (R5),R3
MOVL    UCB$Q_FR4 (R5),R4
JSB     @UCB$L_FPC(R5)
```

The routine that is invoked by this method resumes the driver by calling the EXE$KP_RESTART routines.

All Step 1 and Step 2 drivers that use the second method need to be changed to either of the following:

- Use the EXE$KP_RESTART routine directly (that is, the first method)
- Invoke the resume routine via the standard call sequence:

```
PUSHL   R5              ;P3 = UCB address
PUSHL   UCB$Q_FR4(R5)   ;P2 = FR4 value
PUSHL   UCB$Q_FR3(R5)   ;P1 = FR3 value
CALLS   #3,@UCB$L_FPC(R5)
```

4.11.8 Restriction on Passing C Structure Member Names 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:

1. You pass one of these simple macro names for a structure member as part of a parameter to a C macro, and
2. 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, or
- Introducing additional temporary storage in the outer macro to avoid passing the input arguments directly to the nested macros

4.11.9 Interface Change to IOC$CRAM_CMD System Routine

V6.1

The Step 1 IOC$CRAM_CMD routine interface, which is described in *Creating a Step 1 Driver from an OpenVMS VAX Device Driver*, is as follows:

```
    status = ioc$cram_cmd (cmd_index,
                           byte_offset,
                           ADP,
                           CRAM,
                           [buffer_ptr]);

Inputs:

    cmd_index   Unsigned longword.
    byte_offset Unsigned longword.
    adp         Signed longword.  Address of ADP.
    cram        Signed longword.  Address of CRAM.
    buffer_ptr  Signed longword.  Optional.  Address of buffer,
                2 quadwords in length.

Outputs:

    SS$_NORMAL  success, CRAM successfully initialized
    SS$_BADPARAM failure.  Bad input argument.
```

The optional buffer_ptr argument was removed in FT3. FT4 contains a new argument called the IOHANDLE, which is a magic number supplied by a platform independent I/O bus mapping routine. The platform independent I/O bus mapping routine (IOC$MAP_IO) is available in FT4. When the IOHANDLE argument is present, IOC$CRAM_CMD will use the IOHANDLE to do its address calculations.

The new IOC$CRAM_CMD interface for OpenVMA AXP Version T6.1-FT4 is as follows:

```
    status = ioc$cram_cmd (cmd_index,
                           byte_offset,
                           ADP,
                           CRAM,
                           [IOHANDLE]);

Inputs:

    cmd_index   Unsigned longword.
    byte_offset Unsigned longword.
    adp         Signed longword.  Address of ADP.
    cram        Signed longword.  Address of CRAM.
    iohandle    Quadword.  Magic number
                returned by platform-independent bus
                mapping routines.
Outputs:

    SS$_NORMAL  success, CRAM successfully initialized
    SS$_BADPARAM failure.  Bad input argument.
```

If you are currently calling IOC$CRAM_CMD with 4 input arguments, you do not have to make any changes. If you are calling IOC$CRAM_CMD with 5 arguments with the 5th argument a zero, you do not have to make any changes. If you are currently calling IOC$CRAM_CMD with 5 arguments with a non-zero 5th argument, you will have to make a change to eliminate the buffer_ptr argument.

For more information about the IOC$CRAM_CMD routine, see *OpenVMS AXP Device Support: Reference.*

4.11.10 New Bus Support Routines for I/O Bus Device Access

V6.1 FT4 contains the following new platform independent mapping and access routines:

- IOC$MAP_IO
- IOC$READ_IO
- IOC$WRITE_IO
- IOC$UNMAP_IO

The IOC$MAP_IO routine maps I/O bus physical address space into an address region accessible by the processor. The IOC$UNMAP_IO routine is provided to unmap a previously mapped space, returning the IOHANDLE and the PTEs to the system. IOC$READ_IO and IOC$WRITE_IO are platform independent I/O access routines that provide a platform independent way to read and write I/O space without the overhead of CRAM allocation and initialization. These routines require that the I/O space that is to be accessed have been previously mapped by a call to IOC$MAP_IO. For more information about these routines, see *OpenVMS AXP Device Support: Developer's Guide.*

4.12 Source Level Debugging Support for Device Drivers

V6.1 OpenVMS AXP Version T6.1-FT4 supports a new programming tool that can be used to debug non-pageable system code and device drivers running at any IPL. The OpenVMS AXP System-Code Debugger (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 following known problems and limitations:

- The performance of the system-code debugger's DECwindows Motif interface is lower than the character cell interface.
- The CALL and SET WATCH commands are not currently implemented.
- 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.
- 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. This is obvious 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, please file a QAR and provide the error message along with the MAP file for the image.

4.13 Executive Notes

The following sections provide information about the executive for OpenVMS AXP.

4.13.1 Executive Functions Not Included in OpenVMS AXP

V1.0 Support for PDP–11 compatibility mode will not be included in this release or in future releases of OpenVMS AXP.

4.13.2 New Condition Code for Cross-Mode Page-Read Errors

V1.5 OpenVMS has historically treated as a special case those page-read errors that occur for a page owned by user or supervisor mode (outer mode) but accessed from executive or kernel mode (inner mode) at the time of the pagefault. In this case, the page that incurs a page-read error would be zeroed, the protection of the page would be set to Exec Write (thereby allowing no access to supervisor or user modes), and the process would be allowed to continue referencing this page. No condition code would be signalled to the process.

The intent of this algorithm is to prevent a system failure that would occur when a system service (executing in inner mode) accesses a page that is not in the working set and that cannot be read from the paging disk. This algorithm assumes that the process will return to user mode and that any subsequent attempt to access the page will result in an access violation, which causes the process to abort. In most cases, this assumption is valid.

However, in some cases, the program continues to access (from inner mode) data in pages that were originally created in outer mode. No access violation occurs as the algorithm above expects. The "hidden" zeroing of pages for which page-read errors occur can lead to eventual undetected corruption of the volatile network database as the page of zeroes continues being processed.

This behavior has changed in Version 1.5. Instead of zeroing the affected page, the new condition SS$_PAGRDERRXM (page-read error across access modes) is signalled. If no established handler fields this exception, the system's last-chance handler aborts (deletes) the process.

4.13.3 Spinlock Changes

V1.0

The rank values for all current static spin locks have been spread out to allow for future additions.

All known references to the JIB dynamic spin lock have been replaced with interlocked code sequences, allowing the spin lock itself to be eliminated.

4.14 F$FILE_ATTRIBUTES Lexical Function Returns Incorrect Value

V1.5

When you use the STORED_SEMANTICS item code with the F$FILE_ATTRIBUTES lexical function, an incorrect ASCII string is returned. Digital expects to fix this problem in a future release of the OpenVMS AXP operating system.

4.15 File System Notes

The following sections provide information about file system support in this release of OpenVMS AXP. All basic disk file system support is present in OpenVMS AXP.

4.15.1 Restrictions

V1.5

The following restrictions apply to file system support in Version 1.5:

- ODS-1 format disks are not supported on OpenVMS AXP.
- The Movefile operation is *not* available. Because of this, do not use online disk defragmenters that rely on the Movefile operation.

4.15.2 File Definition Language

V1.0

The File Definition Language (FDL) facility is included in this version of OpenVMS AXP. FDL provides FDLSHR.EXE and CREATEFDL.EXE, the latter invoked with CREATE/FDL.

In addition, OpenVMS AXP includes the Edit/FDL (EDIT/FDL) utility. You can use EDIT/FDL to create new files, but you may not be able to edit existing files.

4.15.3 National Character Set

V1.0 The National Character Set (NCS) runs as a native AXP image. All user-visible functions and interfaces of NCS are identical to OpenVMS VAX NCS. An NCS library created on a VAX system is supported on an AXP system.

4.16 Full IEEE Floating-Point Support Available

V6.1 OpenVMS AXP Version T6.1-FT4 provides support for IEEE Standard for Binary Floating-Point Arithmetic (IEEE Std 754-1985). The OpenVMS AXP IEEE floating-point support provides the following features:

- Support for IEEE trap behavior.

  IEEE Standard 765-1985 specifies five types of exceptions (Invalid Operation, Division by Zero, Overflow, Underflow, and Inexact), and the behavior when such exceptions occur. IEEE Standardd 765-1985 also specifies that the user should be able to request a trap on any of the five exceptions by specifying a handler for it.

  OpenVMS AXP Version T6.1-FT4 provides support that is compliant with the IEEE trap behavior.

- Support for IEEE non-finite arithmetic.

  The Alpha Architecture supports non-finite IEEE arithmetic operations (involving infinity operands, Nans, and denormals) by reporting an arithmetic trap. The Alpha Architecture specifies that a software completion handler interposed between the hardware and the user application will provide the correct behavior.

  OpenVMS AXP Version T6.1-FT4 includes an IEEE completion handler that emulates the IEEE non-finite arithmetic operations in software.

For more information on IEE floating-point support, see the *OpenVMS Programming Concepts Manual*.

4.17 Image Activator Length Comparison Problem Fixed

V6.1 In rare circumstances prior to OpenVMS AXP Version T6.1-FT4, you could build a correct image with the OpenVMS linker and receive the following error message when you tried to run the image:

```
IMGACT-F-BAD_FIXUPVEC, the fixup vector contains inconsistent data
```

The message occurred because a length comparison in the image activator was off by 1 byte. The failure happened when the necessary fixup information exactly filled the area of the image allocated for that purpose.

This problem has been fixed in OpenVMS AXP Version T6.1-FT4.

4.18 Effects of a Failure During an I/O Write Operation

V1.5 The 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. Applications that must guarantee the successful completion of a write operation can verify that the data is on the media by specifying the data check function modifier IO$M_DATACHECK, which is described in the *OpenVMS I/O User's Reference Manual*. Note that the IO$M_DATACHECK data check function, which compares the data in memory with the data on disk, affects performance because the function incurs the overhead of an additional read operation to the media.

If a system failure occurs while a multiple-block write operation is in progress, the operating system does not guarantee the successful completion of the write operation. (OpenVMS does guarantee single-block write operations to DSA drives.) When a failure interrupts a write operation, the data may 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 write operation in progress and the remainder of the blocks contain the data that was present before the write operation.
- The new data was never written to the disk blocks on the media.

To guarantee that a write operation either finishes successfully or (in the event of failure) is redone or rolled back as if it were never started, use additional techniques to assure data correctness and recovery. For example, using database journaling and recovery techniques allows applications to automatically recover from failures such as the following:

- 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 or controller failure or a failure of node-to-node communications.
- Failure of a CPU (such as a system failure, system halt, power failure, or system shutdown) during a multiple-block write operation.
- Mistaken deletion of a file.
- Corruption of file system pointers.
- File corruption due to a software error or incomplete bucket write operation to an indexed file.
- Cancellation of an in-progress multiple-block write operation.

4.19 LIB$FIND_IMAGE_SYMBOL Routine

V1.5 The LIB$FIND_IMAGE_SYMBOL routine sometimes uses an incorrect value as the base address of an image's symbol vector, resulting in an access violation. This occurs when both of the following conditions are met:

- The image being referenced by LIB$FIND_IMAGE_SYMBOL was installed using the /RESIDENT qualifier.
- The image being referenced by LIB$FIND_IMAGE_SYMBOL was already activated, and the initial activation of the image was not performed by LIB$FIND_IMAGE_SYMBOL. (That is, activation was done by direct call to SYS$IMGACT.)

Use either of the following methods to work around this problem:

- Reinstall the image being referenced, but do not specify the /RESIDENT qualifier.

- Activate the uninstalled copy of the image, not the installed copy. To do this, define a logical name for the image being referenced, as shown in the following example for the DPML$SHR image:

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

  Note that you must include a semicolon after the file type in the file specification.

4.20 LIBRARIAN Operation Failure

V1.5 The OpenVMS AXP 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.

4.21 Linker Utility

The following sections apply to the OpenVMS AXP Linker utility (linker).

4.21.1 Suggestion for Improving Application Code Performance

V1.5 You can improve the performance of your application code by taking advantage of the granularity-hints region (thereby improving utilization of the translation buffer). To do this, link the program with the /SECTION_BINDING switch and install the program with the /RESIDENT switch. For more information, see the *OpenVMS Linker Utility Manual* and the *OpenVMS System Manager's Manual.*

4.21.2 Restrictions

V1.0 The following restrictions apply to the linker:

- 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 globals 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*.

- The OpenVMS AXP Version 1.0 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 qualifer /EXTERN=COMMON.

4.22 Lock Manager Synchronization Changes

V6.1 The synchronization for the OpenVMS Lock Manager has changed with this release of OpenVMS AXP. A new spinlock with a name of LCKMGR is now used to synchronize the OpenVMS Lock Manager for standalone OpenVMS AXP machines. AXP systems running as part of a VMScluster still synchronize with the SCS (IOLOCK8) spinlock. Use of the LCKMGR spinlock in a VMScluster will correctly lock the SCS spinlock.

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 use of the LCKMGR spinlock. This change does not affect software on OpenVMS VAX.

4.23 LTDRIVER Restriction

V6.1 LTDRIVER did not set the "extended DDT" bit so 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 cancelled, any $QIO done to the port driver (i.e. with the IO$_TTY_PORT function modifier—like a LAT connect $QIO) *cannot* be cancelled with CANCEL SELECTIVE. This problem will be addressed in a future OpenVMS release.

4.24 MACRO-32 Compiler for OpenVMS AXP

The following sections contain information pertaining to the MACRO-32 Compiler.

4.24.1 MACRO-32 Compiler Now Native

V6.1 The MACRO-32 compiler executable image is now native rather than translated, which should result in noticeable performance improvements.

4.24.2 New Function

V1.5 When /FLAGGING=INSTRUCTIONS is enabled, absolute addresses detected by the compiler are flagged. This function is not documented in *Migrating to an OpenVMS AXP System: Porting VAX MACRO Code*. For example, MOVL R0, 200 compiles correctly (updating memory location 200), but the desired absolute address may be different on an AXP computer. As a result, the informational message CHKABSADR is reported. The /NOFLAG=INSTRUCTIONS qualifier prevents these informational messages from being reported.

4.24.3 Restrictions and Known Problem

V1.5 The following restrictions exist in this version:

- For macros that are read from libraries (.MLBs) at compile time, the debugger reports "Unable to open source file *<library name>*," and the macro source is not shown. This includes the entry point macros, such as .JSB_ENTRY and .CALL_ENTRY, whose definitions are read from STARLET.MLB. In a future release, the compiler will not direct the debugger to read library source files.
- When debugging code, if you examine data declared with .ASCIC or .ASCIZ, it is examined as .ASCII data. That is, the count (for ASCIC) or the null terminator (for ASCIZ) is interpreted as character data.

  To work around this, explicitly specify EXAMINE/ASCIC or EXAMINE/ASCIZ when you examine the data or DEPOSIT/ASCIC or DEPOSIT/ASCIZ when you change the data.
- The MACRO-32 compiler does not support the creation of separate object files from source files separated by a comma (,).
- 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.
- *V6.1* For procedures declared using the .JSB_ENTRY directive, the MACRO-32 compiler will automatically generate 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. For more information about this debugger behavior, see Section 4.7.2.15.

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.

The following problem exists in this version:

> INSV instructions do not generate code that correctly preserves granularity when granularity preservation is turned on.

See Section 5.5 for information about corrections that apply to the *Migrating to an OpenVMS AXP System: Porting VAX MACRO Code*.

4.25 Data Size Returned by Callable MAIL User Routines Corrected

V6.1

Programs that call the MAIL$USER_BEGIN and MAIL$USER_GET_ INFO user routines can request that these routines use a 16-bit word to store the length (in bytes) of returned information at an address that the calling program specifies in the output item list.

Prior to OpenVMS AXP Version T6.1-FT4, these user routines were encoding the length of the returned information in a longword (32 bits) rather than a word (16 bits), thus overwriting space adjacent to the specified address.

The problem has been corrected by ensuring that these routines encode the length of the returned information in a word.

4.26 Programming Consideration for Manipulation of Absolute Queues

V1.0

Some programming languages, such as BLISS, manipulate absolute queues using queuing instructions hidden within a macro definition found in STARLET. These macro operations are not atomic. For example, the BLISS REMQUE macro can be interrupted between the instruction that loads the entry to be removed and the instruction that removes the entry from the queue, as shown:

```
LDL     R16, n(Rn)       ; pointer to element in queue
REMQUEL                  ; remove element from queue
```

This is most notable when a specific absolute queue manipulation occurs at a level other than AST but gets interrupted by an AST routine also manipulating the same absolute queue. The appropriate instruction to use in this case is REMQUEL/D, as shown in the following example:

```
MOV     Rn, R16          ; address of pointer to element in queue
REMQUEL/D                ; remove element from queue
```

This instruction performs the same operation as REMQUEL but is addressed by the longword addressed by R16. This guarantees behavior like that found on VAX systems.

See the *Alpha Architecture Reference Manual* for further details about deferred queue instructions.

4.27 RMS$_NETBTS Status Replaces RMS$_RTB and RMS$_RSZ for Remote Files

V6.1

Prior to OpenVMS AXP Version T6.1-FT4, if you attempted to retrieve or insert /update a record, the following errors would be returned if any buffer was too small for the record:

Operation	Message
$GET (or DCL READ)	%RMS-W-RTB, nnn byte record is too large for users buffer
$PUT or $UPDATE (or DCL WRITE or WRITE /UPDATE)	%RMS-F-RSZ, invalid record size

As of OpenVMS AXP Version T6.1-FT4, these errors are returned only if the buffer that is too small is one of the user record buffers (pointed to by either RAB$L_UBF or RAB$L_RBF) allocated in the user application. If the file is a remote file and the buffer that is too small is one for which the size is determined by the network block count (NBC), then the new RMS-E-NETBTS (network buffer too small) error status is returned. User action assistance for responding to this error is provided in the HELP message database.

4.28 Run-Time Libraries

The following sections apply to run-time libraries. Also refer to Section 4.4 for information about the run-time system support provided for DEC COBOL Version 1.0.

4.28.1 Run-Time Libraries Not Included

V1.5

The run-time libraries listed in Table 4–3 are not included in this version of OpenVMS AXP.

Table 4–3 Run-Time Libraries Not Included

DBGSSISHR	DEBUG item, not required in OpenVMS AXP (replaced by SYS$SSISHR)
DNS$RTL	No DNS in OpenVMS AXP
DNS$SHARE	No DNS in OpenVMS AXP
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 5.4 are available in this version of OpenVMS AXP. The OpenVMS VAX Version 5.4 libraries that are not available are either not being ported to OpenVMS AXP or are planned for a later release of OpenVMS AXP.

For example, the vector math libraries VBLAS1RTL and VMTHRTL are not available in OpenVMS AXP because there is no support on OpenVMS AXP for programs that use the VAX vector instructions.

4.28.2 Native DEC C Run-Time Library Notes

V1.5 This section describes the changes, limitations, and known problems with the OpenVMS AXP Version 1.5 DEC C Run-Time Library (DEC C RTL). Many of these release notes originally appeared in the DEC C for OpenVMS AXP compiler release notes.

4.28.2.1 Incompatibilities Between Version 1.5 and OpenVMS AXP Version 1.0

V1.5 This section describes the incompatibilities between the DEC C RTL for OpenVMS AXP Version 1.5 and the DEC C RTL for OpenVMS AXP Version 1.0.

Several errno constant values have been changed. The change was made for compatibility with OpenVMS VAX Version 6.0, ULTRIX, OSF/1, and the TCP/IP for OpenVMS products. The following errno constants are affected: EDEADLK, ENOTEMPTY, EIDRM, ENOSYS, ENOMSG, ENOLCK, ECANCELED, EFAIL, ENOTSUP, ENWAIT, EINPROG, and EFTYPE. The corresponding errno.h is available with the DEC C Version 1.3 compiler.

The size of the DEC C RTL internal buffer has been increased from 16 kilobytes to 32 kilobytes. The value of the BUFSIZ constant has been modified to correspond to this change. Programs that allocate a buffer of size BUFSIZ (where BUFSIZ is less than 32 KB) and then call setbuf( ) to cause the C RTL to use the new buffer should be rebuilt using a stdio.h that has the BUFSIZ constant set to 32768. Failure to rebuild the source may result in stack corruption because the C RTL might write past the end of the buffer. The official modified stdio.h is available with the DEC C Version 1.3 compiler. See Section 4.28.2.8 for additional details.

4.28.2.2 New Features

V1.5 The following sections describe the new functionality in the OpenVMS AXP Version 1.5 DEC C RTL.

4.28.2.2.1 New Curses Implementation A new implementation of curses (one that is more compatible with UNIX operating systems) is contained in this release of the DEC C RTL. The new curses implementation fixes many problems reported against the OpenVMS curses implementation and is generally more efficient. The new curses.h, which is required to use the new curses implementation, is provided with the DEC C Version 1.3 compiler.

Existing programs are not affected by the new curses functions, because all of the existing OpenVMS curses functions are still available. However, new curses programs built with the DEC C Version 1.3 compiler use the new curses functions by default. To continue to use the OpenVMS curses implementation with the DEC C Version 1.3 compiler, you must compile those modules that include curses.h with the following qualifier:

```
/DEFINE=(__VMS_CURSES)
```

The new curses functions do not provide the support required to call SMG$ routines with the pasteboard and keyboard allocated by the curses functions. Consequently, curses programs that rely on calling SMG$ entry points, as well as curses functions, must continue to use the OpenVMS curses implementation.

The new curses implementation is not interoperable with the old implementation. Attempts to mix calls to the new functions and the old functions result in incorrect output displayed on the screen and may result in an exception from SMG.

4.28.2.2.2 New Single-Character I/O Macro Support Support for macro versions of single-character I/O functions (such as getchar and putchar) is included in this release of the DEC C RTL. The macro versions of these functions are significantly more efficient than the routine versions because the routine call overhead is reduced. The new stdio.h required to use the new macros is provided with the DEC C Version 1.3 compiler.

Existing programs that call single-character I/O functions are not affected by this change. However, some programs that call single-character I/O functions and are built with the DEC C Version 1.3 compiler may encounter problems using the macros. Programs that access record format files are most likely to encounter problems. To continue to use the routines, you must modify your code to add the following statements after you include stdio.h:

```
#undef getchar
#undef putchar
#undef fgetc
#undef fputc
#undef getc
#undef putc
```

4.28.2.3 Restrictions in the DEC C RTL

The following sections document known restrictions in this release of the DEC C RTL.

V6.1 **4.28.2.3.1 DEC C RTL Called from User Mode** The standard DEC C RTL does not support being called from a privileged mode. While certain functions work when called from a privileged mode (for example, the string routines), most of the C RTL functions do not.

In OpenVMS AXP Version T6.1-FT4, a subset of these functions can be called from inner modes by linking with the /SYSEXE qualifier. For more information, see the *OpenVMS AXP Version 6.1 New Features Manual.*

V1.5 **4.28.2.3.2 Modifying Constants Passed by Reference Can Cause an Access Violation** In some compilation modes, the DEC C for OpenVMS AXP compiler allocates string constants in a read-only program section (PSECT), so that constants passed to run-time library routines by reference may cause an access violation (ACCVIO). ACCVIOs will occur only if the RTL routine attempts to modify the argument. For example, the following routine call causes an access violation:

```
mktemp ("namXXXXXX")
```

The access violation occurs because the DEC C compiler places pointers to constants in a read-only program section (PSECT). By placing pointers to constants in read-only PSECTs, run-time library modifications to constant arguments are prevented.

This treatment of constants also applies to constructs of the form “&0”, that is, passing a pointer to zero. Examine all such uses to prevent existing code from causing access violations.

V1.5 **4.28.2.3.3 Translated VAX C RTL Interoperability Restrictions** The native DEC C RTL does not interoperate with the translated VAX C RTL. Each C RTL has its own context, which means that files, signals, or any other context information cannot be shared.

This also includes vfork/execute (exec( ) function) support. The DEC C RTL is unable to execute a program linked with the translated VAX C RTL. Likewise, the translated VAX C RTL cannot execute a program linked with the native DEC C RTL.

4.28.2.4 DEC C RTL Images

V1.0 The images that comprise the OpenVMS AXP Version 1.5 DEC C RTL are different from the OpenVMS VAX Version 5.4 VAX C RTL images in the following ways:

- The DEC C RTL for OpenVMS AXP Version 1.5 does not include the VAXCRTL.EXE or VAXCRTLG.EXE shareable images. Instead, the image DECC$SHR.EXE (which resides in IMAGELIB.OLB) is used. This image contains all DEC C RTL functions and data and has a namespace that conforms with OpenVMS standards. (All external names are prefixed with "decc$".) To make use of this image, all C RTL references must be prefixed with "decc$" so that the proper code in DECC$SHR.EXE is accessed.

  This prefixing occurs as the default action of the DEC C compiler. Refer to the compiler documentation on the /PREFIX qualifier for more information.

  Because DECC$SHR is in IMAGELIB.OLB, you need not refer to DECC$SHR in any linker options files. The linker automatically searches IMAGELIB for unresolved symbols.

- The contents and names of the object libraries have changed. The actual code for the object libraries has also been changed. (All external names are prefixed with "decc$".) This code has been placed in STARLET.OLB. The object libraries VAXCRTL.OLB, VAXCRTLD.OLB, and VAXCRTLT.OLB contain the unprefixed entry points for G_floating, D_floating, and IEEE_floating versions of each RTL function, respectively. Each entry point simply transfers control to the proper code in IMAGELIB.OLB or STARLET.OLB. If the compiler has prefixed RTL function names, there is no reason to use the VAXCRTL object libraries. However, if prefixing by the compiler is turned off, one of the object libraries must be used.

  When you link C programs on OpenVMS AXP Version 1.5, the linker uses the C RTL shareable image by default. In order to link against the C RTL object library instead of the shareable image, you must use the /NOSYSSHR qualifier on the link command.

  Note that linking against one of the VAXCRTL object libraries finds the C RTL symbols in DECC$SHR and does not include the C RTL code in your image. You must specify the /NOSYSSHR qualifier to use the C RTL symbols in STARLET and to eliminate the dependence of your program on the DECC$SHR shareable image.

  Programs that use the C RTL curses or socket functionality, or the DECthreads reentrancy mode, should not link /NOSYSSHR. The routine LIB$FIND_IMAGE_SYMBOL, used by those sets of functionality, is not in STARLET.OLB. Note that the C RTL references to LIB$FIND_IMAGE_SYMBOL have been changed. As a result, the linker does not issue any warning messages when you link programs using /NOSYSSHR. However, an attempt to use curses, sockets, or the DECthreads reentrancy mode by a program linked /NOSYSSHR results in an access violation (ACCVIO).

4.28.2.5 Incompatibilities Between DEC C RTL and VAX C RTL

V1.0

The following list describes functional incompatibilities between the OpenVMS AXP Version 1.5 DEC C RTL and the OpenVMS VAX Version 5.4 VAX C RTL. These changes are, for the most part, a result of making the DEC C RTL compliant with ANSI C.

- The vfork and setjmp functions are not implemented as actual functions in the OpenVMS AXP Version 1.5 DEC C RTL but are implemented as macros. Any program referring to those entry points must include the appropriate header file, or the linker reports vfork or setjmp, or both, as being undefined.
- Exit handlers established by atexit( ) are no longer executed when abort( ) is called.
- Files opened in append mode always have writes forced to the end of the file. With the VAX C RTL, the behavior of files opened in append mode was no different than for files opened in write mode.
- File positioning (such as fseek( )) is now allowed on files opened in append mode. Note that all writes are still forced to the end of the file.
- A call to ungetc( ) clears the EOF flag for the file.
- A call to a file-position routine, such as fseek( ), clears the EOF flag for the file.
- The EOF flag for a file is always cleared when the file is first opened. The VAX C RTL sets the EOF flag to true if the opened file is empty.
- A call to a file-position routine (such as fseek( )) loses the character pushed back by a call to ungetc( ).
- The C signal SIGABRT is equal to the SIGIOT signal rather than to SIGILL, as in previous versions. Programs fail if they look for the exception SS$_OPCDEC in order to prevent an abort( ) from occurring. Programs should now look for the exception SS$_OPCCUS.
- Calling setvbuf( ) with a buffer smaller than BUFSIZ bytes results in the user's buffer not being used for most files.
- The signal( ) function returns SIG_ERR on error rather than the -1 of previous versions. This should not affect the behavior of programs that check for -1.
- The gets( ) and fgets( ) functions no longer place a null character at the beginning of the input buffer when EOF is reached and no characters have been read. Programs that assume the input buffer to be empty when a null pointer is returned from the function do not behave correctly.
- The following SHELL$ routines from the VAX C RTL have been replaced with DECC$ equivalents in the DEC C RTL (for example, decc$to_vms replaces shell$to_vms). Please see the unixlib.h header file for prototypes.
  - shell$fix_time
  - shell$from_vms
  - shell$match_wild
  - shell$to_vms
  - shell$translate_vms

- The ANSI padding rules for the printf( ) routines differ from the VAX C RTL behavior. See Table 4–4 for examples of the new and old behavior.
- The ANSI rules for the output of floating-point numbers differ from the VAX C RTL behavior, so that a digit of precision might be lost, depending on the format specifier and precision given. See Table 4–4 for examples of the new and old behavior.

Table 4–4 Changes in Printf( ) Behavior Required by the ANSI C Standard

Example	VAX C RTL	DEC C RTL
printf("%04.2d", 77)	"0077"	" 77"
printf("%6.4d", 77)	" 77"	" 0077"
printf("%.2g", 9.876e+2)	"9.88e+02"	"9.9e+02"

4.28.2.6 Obsolete Functionality

V1.5

The following routines are considered obsolete in the DEC C RTL and should not be used in new development. Digital will continue to provide them for compatibility with the VAX C RTL. The DEC C RTL versions of the standard C memory allocation routines are faster and more efficient than those in the VAX C RTL.

- VAXC$CALLOC_OPT
- VAXC$CFREE_OPT
- VAXC$FREE_OPT
- VAXC$MALLOC_OPT
- VAXC$REALLOC_OPT

The global symbols sys_nerr and sys_errlist are obsolete. Use the functions perror( ) and strerror( ) to obtain the information previously provided by sys_nerr and sys_errlist.

The following SHELL$ routines from the VAX C RTL are not available in the OpenVMS AXP Version 1.5 DEC C RTL:

- shell$cli_name
- shell$get_argv
- shell$is_shell

The OpenVMS curses routines and macros are considered obsolete with this release of the DEC C RTL and if possible should not be used in new development. Digital will continue to provide them for compatibility with the VAX C RTL and for the OpenVMS AXP Version 1.0 DEC C RTL. The new curses implementation (Section 4.28.2.2.1) is more compatible with UNIX systems and fixes many of the problems reported against the OpenVMS curses functions.

4.28.2.7 Enhancements Not Found in the VAX C RTL

V1.0

The following list describes enhancements that have been made to the OpenVMS AXP Version 1.5 DEC C RTL. These enhancements are not included in the OpenVMS VAX Version 5.4 VAX C RTL:

- The assert macro in assert.h now conforms to the ANSI C standard. This change does not affect the behavior of assert.
- Seeks on a fixed-length record file are made to the exact byte specified rather than to record boundaries only. Note that this is true only for fixed-length record files. Seeks done on variable-length or VFC record files are still done only to record boundaries.
- If the parameter to fflush is NULL, then all buffers associated with all currently open files are flushed.
- File positioning, such as fseek( ), is allowed on files opened in append mode.
- The ftell( ) function reports a file position that takes into account the character pushed back by a call to ungetc( ).
- The character classification functions, such as isupper( ), isalpha( ), and so forth, have been extended to handle characters in the range from 128 to 255.
- Many routines that were provided only as C macros in the OpenVMS VAX Version 5.4 VAX C RTL, such as the is* routines, clearerr, feof, ferror, and fileno, are now provided as both C macros and as functions.
- The formatted input functions, such as scanf and fscanf, return 0 on early matching failure.
- The VT400 class of terminals is now recognized by getenv("TERM").
- Most routines have been made reentrant. For more information, see the section titled Specific Portability Concerns and the information about the new function DECC$SET_REENTRANCY in the *DEC C Run-Time Library Reference Manual for OpenVMS Systems*. Also see the compiler documentation for the switch /REENTRANCY.

4.28.2.8 Performance Considerations

V1.5

This section describes performance features of the OpenVMS AXP Version 1.5 DEC C RTL.

- The OpenVMS AXP Version 1.5 DEC C RTL provides an alternate implementation of setjmp that is faster than the default implementation. To use the faster implementation, compile any source that invokes the setjmp function using the switch /DEFINE=__UNIX_SETJMP. See the setjmp.h header file for additional details. The faster implementation achieves its speed by not calling any OpenVMS condition handlers in the call stack. If your program relies on OpenVMS condition handlers being called during an unwind, you cannot use the faster setjmp implementation. This alternate setjmp implementation does not conform to the *OpenVMS Calling Standard*.
- The OpenVMS AXP Version 1.5 DEC C RTL is now reentrant (see Section 4.28.2.7 for more details). This reentrancy is achieved at a performance cost. If your program does not require reentrancy in the DEC C RTL, you can eliminate the reentrancy performance penalty. To do so, compile the module containing your main routine with the switch /REENTRANCY=NONE, or call DECC$SET_REENTRANCY(C$C_NONE) in your main routine. However, programs that use DECthreads, call C RTL

routines from within condition handlers or signal handlers, use asynchronous system traps, or implement their own thread mechanisms probably will not function correctly if DEC C RTL reentrancy is disabled.

- In stream access mode, which is emulated using RMS block mode, the default block size is 64 rather than the VAX C RTL value of 1. The constant BUFSIZ has been changed accordingly to 32768 (number of blocks multiplied by the block size of 512). This change results in significantly improved stream I/O performance. Note, however, that programs that specify an "mbc=" clause in their file open call may reduce default performance if the mbc value is less than 64.
- A problem in the C RTL which slowed record I/O operations has been fixed. The C RTL record I/O performance is now equivalent to that of the VAX C RTL.

4.28.2.9 Known Problems

V1.5 This section describes existing problems that have been reported against the DEC C RTL that were not fixed for OpenVMS AXP Version 1.5.

- File output may not produce correct results if the process's file quota has been exceeded.
- The malloc entry point is missing from the VAXCRTL* object libraries.
- The getenv( ) function receives an access violation if the symbol being requested does not exist.
- The printf( ) function receives an access violation if no virtual memory is available.
- The chdir( ) function does not permanently change the default directory when the program is in executive mode. This functionality has not yet been implemented for OpenVMS AXP.
- The nice( ) function returns a success status when given an out-of-range priority value.
- The nice( ) function does not support priority levels above 32.
- The sbrk( ) function does not detect that no virtual memory is available.
- The fcvt( ) function returns a NULL pointer when the number of fractional digits requested is zero. Use the following workaround:

  ```
  sprintf( buffer, "%.0f", data );
  ```
- The C RTL returns -HUGE_VAL from the exp( ) function for the overflow condition, instead of HUGE_VAL.
- A call to fgetname( ) with a null filename does not return the current device and directory.
- The C RTL currently limits the number of simultaneously open files to 256.
- The CPU time consumed by a system( ) call is not included in the time returned by the clock( ) and times( ) functions.
- The C RTL currently does not set the longest record length value for a file that is created using stream I/O.
- The parent process does not detect the disappearance of a child created with the vfork( ) function.

- The C RTL hangs if a buffer larger than 65535 bytes is passed to the setvbuf( ) function.

4.28.3 Compatibility Between the VAX and AXP Mathematics Libraries

V1.0 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 AXP system. Jacket routines are provided that map MTH$ routines to their math$ counterparts in the Digital Portable Mathematics Library (DPML) for OpenVMS AXP. However, there is no support in the DPML for calls made to JSB entry points and vector routines. Please 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.

4.29 Security Changes

V6.1 OpenVMS Version T6.1-FT4 offers significant enhancements to system security and some of these changes can impact your everyday operations. For this reason, take special note of the changes identified in Section 3.35.1.

4.29.1 Site–Specific Hash Algorithm Change

V6.1 Two changes were made 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. However, unless modifications were made to the Digital-supplied preamble code in the example, you need not be concerned with the following functional changes. As before, site-specific algorithm dispatching generally only replaces a "NOP" instruction in the template.

First, 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 only have to recompile once.

Second, 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.

4.30 STARLET Data Structures and Definitions for C Programmers

V1.0 OpenVMS AXP 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 AXP 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 AXP Version 1.0 and 1.5 (not in the new SYS$STARLET_C.TLB). In OpenVMS AXP Version T6.1-FT4, the file SYS$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 to access the .H files

  During installation of OpenVMS AXP 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.

4.31 System Dump Analyzer Utility (SDA) Notes

The following sections describe an enhancement, some problems that have been fixed, and a known problem that affect the System Dump Analyzer utility (SDA).

4.31.1 SHOW MACHINE_CHECK Command Available

V1.0

The SDA SHOW MACHINE_CHECK display works for the DEC 4000 AXP and DEC 7000 Model 600 AXP platforms only. Note that the SHOW MACHINE_CHECK command displays available information even if the cause of the system failure was something other than a machine check. The data is guaranteed to be accurate only if the bugcheck code for the failure is a machine check.

The following display applies to a DEC 7000 Model 600 AXP machine check:

```
Processor specific information:
-------------------------------
Exception address:     00000000 00030122    Exception Summary:   00000000 00000000
Pal base address:      00000000 00008000    Exception Mask:      00000000 00000000
HW Interrupt Request:  00000000 00000342    HW Interrupt Ena:    00000001 FFFFDCE0
MM_CSR                 00000000 00005120    ICCSR:               00000003 F81F0000
D-cache address:       00000007 FFFFFFFF    D-cache status:      00000000 000002E0
BIU status:            00000000 00000041    BIU address [7..0]:  00000003 F8400000
BIU control:           00000008 50006447    Fill Address:        00000000 00006120
Single-bit syndrome:   00000000 00000000    Processor mchck VA:  00000000 00006190
A-box control:         00000000 0000040E    B-cache TAG:         0000609A 30482000

System specific information:
----------------------------
Garbage bus info:      00300009 000000B0    Device type:                  00008001
LCNR:                           00000001    Memory error:                 00000000
LBER:                           00001081    Bus error synd 0,1:  0000000C 0000000C
Bus error cmd:         00000020 00420000    Bus error synd 2,3:  0000000C 0000000C
LEP mode:                       00010010    LEP lock address:             00091B86
```

4.31.2 Known Problems

V1.5

The following problems exist in this version of SDA:

- When you enable full console messages (by setting the DUMPSTYLE system parameter to 0 or 1) and an error halt restart bugcheck occurs (such as kernel stack not valid), the bugcheck code causes an access violation while trying to print out stack information on the console.

  As a result, a recursive bugcheck message is printed on the console. However, SDA continues to write the dump file based on the original crash. The dump file is created as if the recursive bugcheck never happenned.

  Digital expects to fix this problem in a future release.

- The REFCNT field in the SHOW PAGE_TABLE and SHOW PROCESS /PROCESS_SECTION displays is too narrow and prevents large values from being printed. The field displays asterisk characters (*) to indicate this problem. In a future release, the REFCNT field will be widened.

- The FLAGS field in the SHOW PROCESS/PHD display displays only a word of data even though a longword of data may be present. This field will be increased for a future release.

- The ADDRESS column for the SHOW PROCESS/PROCESS_SECTION display may contain the wrong value for P1 space sections. To determine if the value is incorrect, compare the value in the ADDRESS column with the "First free P0 address" field of the SHOW PROCESS/PHD display. If the value in the ADDRESS column is greater, it is a P1 space section.

  To calculate the correct address, enter the following command, where *n* represents the address displayed in the ADDRESS field:

  ```
  SDA> EVAL (((n/2000)-((@sgn$gl_ptpagcnt)@a))*2000) & #80000000
  ```

- The SHOW POOL/NONPAGED command does not work if nonpaged pool is exhausted (that is, when no memory is available). Instead, SDA immediately returns the following error:

  ```
  %SDA-E-NOREAD, unable to access location 00000000
  ```

  There are no workarounds.

- EXAMINE 0:FFFFFFFF does not work. You cannot examine a range that crosses from P0 to P1 space or from P1 to system space. You must examine the ranges separately, as in the following example:

  ```
  SDA> EXAMINE 0:3FFFFFFF
  SDA> EXAMINE 40000000:7FFFFFFF
  SDA> EXAMINE 80000000:FFFFFFFF
  ```

- If an access violation occurs during a bugcheck, system space may not be saved. When this happens, SDA cannot analyze the dump. However, instead of returning meaningful information, SDA returns the following error message:

  ```
  %LIB-F-BADBLOSIZ, bad block size
  ```

4.32 System Services Notes

The following sections describe changes and a restriction that apply to system services in OpenVMS AXP.

4.32.1 Low Four Bits of Chan Argument Are Checked — Change

V1.0

The **chan** argument is used in a number of system services to pass a channel number. Historically, the operating system has always regarded the **chan** argument as a 16 bit unsigned number but has never used the low four bits of the argument. Thus, it is possible that some user programs might use the low four bits of the argument as temporary storage unrelated to the channel number.

Now, the low four bits of the **chan** argument are checked. If any bit is nonzero, a status code of SS$_IVCHAN is returned to the caller of the service. Modify your source code to properly store user information in a location other than the **chan** argument.

4.32.2 Condition Code SS$_PAGRDERR Changed

V6.1

The signal argument of the SS$_PAGRDERR condition has been changed. Prior to OpenVMS AXP Version T6.1-FT4, the first signal argument was the mask of the transition not valid reason. In OpenVMS AXP Version T6.1-FT4, this argument has been changed to provide the actual I/O failure status that caused the page read error.

SS$_PAGRDERR	Type:	Fault
	Description:	Read error occured during an attempt to read a faulted page from disk
	Arguments:	1 I/O failure status
		2 Virtual address of referenced page

4.32.3 Behavior of $CRMPSC and $MGBLSC System Services Differs on AXP and VAX Systems

V1.5

The behavior of the **inadr** argument to the $CRMPSC and $MGBLSC system services is different on AXP and VAX systems.

For more information about the differences, see the *OpenVMS System Services Reference Manual.*

4.32.4 $DCLCMH System Service Behavior Differs on AXP and VAX Systems

V1.0

The Declare Change Mode or Compatibility Mode Handler service specifies the address of a routine to receive control when a Change Mode to User or Change Mode to Supervisor instruction trap occurs or when a compatibility mode fault occurs on a VAX system. You declare the change mode handler from the access mode at which the request is to be handled; then when the process executing in user mode issues a Change Mode to Supervisor instruction, the change mode handler receives control and executes in supervisor mode.

On AXP systems, the change mode dispatcher calls the change mode handler as a normal procedure (that is, with a standard procedure call). The change mode handler must exit by performing a standard procedure return to the change mode dispatcher.

Argument usage (for example, the change mode code) passed between the routine that issued the change mode instruction and the change mode handler is coordinated strictly by agreement between the two procedures, rather than in accord with the *OpenVMS Calling Standard.*

The following MACRO code example shows a subroutine, calling Change Mode to User, written for AXP users porting from VAX systems:

```
CHG_MD: .CALL_ENTRY
        CHMU
        RET
```

Call this subroutine from any program that requires a Change Mode to User instruction to be invoked.

On a VAX system, the top longword of the stack contains the zero-extended change mode code. The change mode handler must exit by removing the change mode code from the stack and issuing an REI instruction.

4.32.5 $FAO System Service

V1.5 The Formatted ASCII Output ($FAO) system service accepts a maximum of 20 directive parameters, in addition to the **ctrstr**, **outlen**, and **outbuf** arguments.

If you specify more than 20 directive parameters, $FAO returns SS$_OVRMAXARG.

4.32.6 $FORMAT_AUDIT Width Argument Does Not Work Consistently — Restriction

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.

4.32.7 $GETJPI System Service — New JPI$_DFMBC Item Code

V1.5 JPI$_DFMBC is a new item code for the $GETJPI system service. When you specify JPI$_DFMBC, $GETJPI returns the default multibuffer count for a process as a longword integer value.

4.32.8 $SUSPND System Service in a Cluster Environment — Incorrect Behavior

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 version of OpenVMS AXP.

4.33 Traceback Handler Support

V1.0 Traceback handler support is present in OpenVMS AXP Version 1.0. The following items describe the available support:

- Symbolic traceback is supported, including traceback for images installed /RESIDENT.

  The symbolic information that is reported includes the image name in which the invocation PC lies, the module name, the routine name, and the line number. Two PC values are also provided:

  - Relative (to the routine base address)

- Absolute

  If the traceback handler must resort to nonsymbolic information (because there is no symbolic information in the image), then the relative PC is relative to the image base address.

- The traceback handler supports the (nonsymbolic) display of translated VAX call frames.
- Exception frame reporting has been implemented.

4.34 Translated Image Environment (TIE) Notes

V1.5 Image translation is one means of migrating all or part of a VAX application to OpenVMS AXP. The VAX Environment Software Translator (VEST) creates a translated image by converting a VAX executable or shareable image into a functionally equivalent AXP image. VEST is a component of the optional layered product DECmigrate for OpenVMS AXP.

When a translated image runs on OpenVMS AXP, 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 performs 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*.

OpenVMS AXP Version 1.5 supports translation of programs from OpenVMS VAX Version 4.0 through Version 5.4-3. Although VEST translates Version 5.5 images, OpenVMS AXP Version 1.5 might not provide the necessary run-time support because the translated libraries are based on Version 5.4-3. In this case, the translated OpenVMS VAX Version 5.5 image may get an ident mismatch at run time.

The following sections discuss these topics:

- Interoperability between native and translated images
- Running translated images
- TIE statistics and feedback
- TIE restrictions

4.34.1 Interoperability

V1.0 The TIE works together with other components of OpenVMS AXP 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 4.34.4. Table 4–5 provides pointers to documentation that describes the procedures.

Table 4–5 Interoperability Documentation

For information about	See
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*

4.34.2 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 will not run correctly unless OpenVMS AXP 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 .IIFs 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 AXP.

OpenVMS AXP includes a set of translated run-time libraries and a matching set of image information files, which are listed in Section 4.35. 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 AXP 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.

Defining Logical Names for Libraries

Note that 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*.

4.34.3 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.

4.34.4 TIE Restrictions

V1.5 The following restrictions apply to the TIE:

Interoperability Restrictions

- A native routine that either calls or is called by a translated image must be compiled with the /TIE qualifier and 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 were not used to compile and link the native routine, an error will occur 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.

  This restriction is permanent.

- 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 suspect 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 AXP 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.

Condition and Exception Handler Restrictions

- There is a 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.

  This restriction is 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 AXP program status (PS) is reported in the signal array instead because there is no VAX PSL.

  This restriction is permanent.

- 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.

This restriction is permanent.

Floating-Point Restrictions

- In some cases, floating-point instructions operating on the same data generate a trap on an AXP system but not on a VAX system. Specifically, VAX floating-point instructions on OpenVMS AXP 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 AXP an application that uses dirty zeros.

  This restriction is permanent.

- 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 following VAX instruction sequence:

  ```
  MOVD    (SP),R2
  MOVD    R2,-(SP)
  ```

  VEST translates these VAX instructions into AXP 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.

  This restriction is permanent.

Translated VAX C Program Restrictions

- 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$EMULAT_TV.EXE, which is dynamically activated via LIB$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.

This restriction is permanent.

- 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.

  This restriction is permanent.

4.34.5 Translated Image Environment Does Not Support Version 6.0 Features

V6.1

System Services and Run-Time Library entrypoints newly added for OpenVMS VAX Version 6.0 are not yet supported with OpenVMS AXP Version T6.1-FT4 field test. This means that VAX programs which use the new Security Services such as $CHECK_PRIVILEGE cannot be translated with VEST (a.k.a. DECmigrate) for use on OpenVMS AXP Version T6.1.

4.35 Translated Images and Related Files

V1.0

This section lists the translated images, image information files, and other related files that are provided with OpenVMS AXP.

OpenVMS AXP contains no translated message images. All message images have been made native.

OpenVMS AXP contains the following translated images in SYS$LIBRARY:

BASRTL2_D53_TV.EXE
BASRTL2_D56_TV.EXE
BASRTL_D56_TV.EXE
BLAS1RTL_D53_TV.EXE
BLAS1RTL_D56_TV.EXE
COBRTL_D56_TV.EXE
DBLRTL_D56_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

OpenVMS AXP provides the following translated images in SYS$SYSTEM:

DBLMSGMGR_TV.EXE
EDF_TV.EXE
MONITOR_TV.EXE
TECO32_TV.EXE

OpenVMS AXP contains the following translated 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
LIBRTL2_D56_TV.EXE
LIBRTL_D56_TV.EXE
PLIRTL_D56_TV.EXE
RPGRTL_TV.EXE
SCNRTL_TV.EXE
TECOSHR_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 4.36 for more information about the translated run-time libraries.

OpenVMS AXP contains the following 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
```

The following system logical names are defined in order 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

4.36 Translated Run-Time Libraries

V1.0

As part of the OpenVMS AXP 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 AXP 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 AXP 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

If you need to use the D56 form of one of these libraries in order to get more consistent results (at a cost of execution time performance), see Section 4.36.1 for instructions.

4.36.1 Accessing the D56 Form of the Run-Time Libraries

V1.0 The following actions happen by default when you use the run-time libraries:

- 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 MTH$ RTL, translated images that invoke MTH$ double-precision floating-point functions get routines that use the D53 data type.
- For all others, the AXP 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 4–6. The logical name can be defined on a per-process or system-wide basis, as appropriate for your site.

Table 4–6 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

4.36.2 Problem with Translated Image Environment

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, AXP JSB semantics are present in the native code (naturally).

To work around this problem, the translated callers need to use CALL instead of JSB.

4.36.3 Translated VAX BASIC Run-Time Library Notes

V1.0 This section describes the limitations and known problems with the OpenVMS AXP translated VAX BASIC run-time library (VAX BASIC RTL). The translated VAX BASIC RTL is a translated version of the OpenVMS VAX Version 5.4 VAX BASIC RTL.

All problems and restrictions present in the OpenVMS VAX Version 5.4 release of the VAX BASIC RTL exist unchanged in the translated VAX BASIC RTL. These include:

- Integer errors do not raise the appropriate exceptions.

  For example, a divide by zero error does not raise a BASIC error handler but causes the translated program to abort. To avoid these problems, explicitly test for integer operand error conditions.

- Array descriptors for H Float data items will not work.

  For example, passing an H float array by descriptor (the default) does not pass the array properly to a function or subprogram. Pass H Float arrays by reference to avoid this problem.

- WHEN block exception processing may not work the same as on VAX systems.

  Translate the program using the /OPTIMIZE=NOSCHEDULE qualifier. Programs containing WHEN blocks handle errors but may dump on exit.
- Explicit calls to LIB$SIGNAL are not handled by a BASIC exception handler in a user program.
- The ON...GOSUB statement does not work properly.

  The control expression is corrupted. Any attempt to access the control variable after the ON...GOSUB statement has finished causes the program to abort.
- Taking the identity of large odd dimension matrices can result in memory management violations.

4.36.4 Translated VAX C Run-Time Library Notes

V1.0

This section describes the limitations and known problems with the OpenVMS AXP translated VAX C Run-Time Library (VAX C RTL).

4.36.4.1 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 some 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.

4.36.4.2 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.

4.37 Translated VAX COBOL Programs Supported

V1.5

The OpenVMS AXP Version 1.5 operating system supports the execution of translated VAX COBOL programs compiled with the VAX COBOL Version 5.0 compiler (or earlier compilers).

Programs compiled with the VAX COBOL Version 5.1 compiler are not supported by the OpenVMS AXP Version 1.5 operating system.

5
Documentation Release Notes

This chapter describes changes to the structure of the OpenVMS documentation set and corrections to published documentation.

5.1 Documentation Changes for OpenVMS AXP Version 1.5

V1.5 Documentation offerings have been expanded for OpenVMS AXP Version 1.5. The major changes are described below. For a complete description of OpenVMS documentation and ordering information, see the *Overview of OpenVMS Documentation.*

5.1.1 Extended Set Redefined

V1.5 With the release of OpenVMS AXP Version 1.5, the name of the Extended Documentation Set has been changed to the Standard Documentation Set. The Standard Documentation Set continues to be a full documentation set with extensive explanatory information on all OpenVMS resources. It is comprised of the following kits (note that the Advanced System Management Kit is new with OpenVMS AXP Version 1.5):

- Base Documentation Kit
- Advanced System Management Kit
- Networking Kit
- Programming Kit
- Master Reference Kit
- AXP Migration Kit

5.1.2 New Standard + Optional Documentation Set

V1.5 We now offer a new documentation set that delivers all optional OpenVMS manuals (except for volume-shadowing and RMS journaling documentation) with the Standard Documentation Set. This new offering is the Standard + Optional Documentation Set. You can also order optional manuals individually. For a complete listing of the contents of the Standard + Optional Documentation Set, see the *Overview of OpenVMS Documentation.*

5.1.3 Three Flavors of Documentation Available

V1.5 All documentation sets are now available in AXP, VAX, or AXP and VAX Combined compilations. For example, the OpenVMS AXP Base Kit contains seven common manuals and the *OpenVMS AXP Guide to System Security*. The OpenVMS VAX Base Kit contains seven common manuals and the *OpenVMS VAX Guide to System Security*. The OpenVMS AXP and VAX Combined Base Kit contains seven common manuals and both security guides. You choose the flavor you need for your particular environment.

5.2 OpenVMS AXP Documentation Set

V1.0 Parts of the OpenVMS AXP documentation set refer to the ALARM_JOURNAL ACE as the ALARM ACE. The ALARM ACE and the ALARM_JOURNAL ACE are synonymous.

5.3 *OpenVMS Debugger Manual*

V1.5 In the Command Dictionary section of the *OpenVMS Debugger Manual*, explanatory text for the /FLOAT qualifier should read as follows:

> On VAX systems, displays each examined entity in the F_floating type (4 bytes). On AXP systems, displays each entity in the IEEE T_floating type (double precision, length 8 bytes).

5.4 *OpenVMS Delta/XDelta Debugger Manual*

V1.5 The base register default offset for OpenVMS AXP is 10000_{16}, not 100000_{16}, as documented in the description of the ;X (Load Base Register) command.

5.5 *Migrating to an OpenVMS AXP System: Porting VAX MACRO Code*

V1.5 The /FLAG=INSTRUCTIONS qualifier is documented as flagging instructions that may compile correctly but that should be examined anyway for correctness on an AXP computer. This applies only to instructions that use absolute addresses and not to the instruction M*x*PR as stated in the description of the /FLAG=INSTRUCTIONS qualifier. If the compiler detects the use of absolute addresses, the instructions are flagged with the informational message CHKABSADR.

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.*

The restriction that pertained to the qualifiers /DEBUG, /DISABLE, and /ENABLE and to the directives .DISABLE and .ENABLE, 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 directive. 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.

5.6 *Migrating to an OpenVMS AXP System: Recompiling and Relinking Applications*

V1.5 *Migrating to an OpenVMS AXP System: Recompiling and Relinking Applications*, the *OpenVMS Linker Utility Manual*, and the description of the BAD_LINK message in the Help Message database in the *OpenVMS System Messages and Recovery Procedures Reference Manual* mistakenly describe /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 AXP images to work with translated VAX images.

5.7 *OpenVMS National Character Set Utility Manual*

V1.0

The command NCS/CREATE/LIBRARY=*filename.ext* defaults to directory SYS$LIBRARY. A device and directory specification is required to create the library elsewhere, as shown in the following example:

NCS/CREATE/LIBRARY=device:[directory]*filename.ext*

The *OpenVMS National Character Set Utility Manual* shows this by example but does not explicitly indicate the default. This information was omitted from the existing documentation.

5.8 Correction to OpenVMS Programming Documentation

V1.0

Appendix A of the OpenVMS VAX Version 5.4 manual *Introduction to VMS System Routines* (currently Appendix B of the OpenVMS AXP manual *OpenVMS Programming Concepts Manual*) describes the OpenVMS usage type *user-arg* as being an "unsigned longword denoting a user-defined argument." This description is correct.

Descriptions of certain system services and run-time library (RTL) routines in OpenVMS programming documentation classify the OpenVMS usage type *user-arg* as being "unspecified". This is incorrect. The user argument parameter to such routines must be specified as a longword containing a nonfloating-point value.

A program that uses the user argument parameter of any of the following routines to pass floating-point data will not run correctly on an AXP system, either as a native AXP image or as a translated VAX image:

- LIB$CREATE_USER_VM_ZONE
- LIB$DECODE_FAULT
- LIB$DELETE_FILE
- LIB$INSERT_TREE
- LIB$LOOKUP_TREE
- LIB$RENAME_FILE
- LIB$SHOW_TIMER
- LIB$SHOW_VM
- LIB$SHOW_VM_ZONE
- LIB$SPAWN

- LIB$STAT_TIMER
- LIB$STAT_VM
- LIB$TRAVERSE_TREE
- SMG$PUT_PASTEBOARD
- SYS$PUTMSG

The current editions of the appropriate OpenVMS programming manuals contain the correct descriptions of user argument parameters to these routines.

5.9 *OpenVMS AXP Version 1.5 Release Notes*

V6.1 Section 4.6.1 of the *OpenVMS AXP Version 1.5 Release Notes* stated that the /EXACT_ORDER qualifier was available for BACKUP. This was an error, but the /EXACT_ORDER qualifier is available for OpenVMS AXP Version T6.1-FT4.

5.10 *OpenVMS RTL Screen Management (SMG$) Manual*

V1.5 In the *OpenVMS RTL Screen Management (SMG$) Manual* for OpenVMS AXP Version 1.5, the Get Pasting Info routine occurs twice. The first instance mistakenly replaced the following Get Pasteboard Attributes routine.

SMG$GET_PASTEBOARD_ATTRIBUTES—Get Pasteboard Attributes

The Get Pasteboard Attributes routine gets pasteboard attributes and stores them in the pasteboard information table.

Format

```
SMG$GET_PASTEBOARD_ATTRIBUTES

                        pasteboard-id, pasteboard-info-table,
                        pasteboard-info-table-size
```

Returns

OpenVMS usage	cond_value
type	longword (unsigned)
access	write only
mechanism	by value

Arguments

pasteboard-id

OpenVMS usage	identifier
type	longword (unsigned)
access	read only
mechanism	by reference

Specifies the pasteboard for which information is requested. The **pasteboard-id** argument is the address of an unsigned longword that contains the pasteboard identifier.

Pasteboard-id is returned by SMG$CREATE_PASTEBOARD.

pasteboard-info-table

OpenVMS usage	unspecified
type	unspecified
access	write only
mechanism	by reference, array reference

Receives the pasteboard attributes. The **pasteboard-info-table** argument is the address of a data structure into which are written the pasteboard attributes.

The values in the **pasteboard-info-table** argument can be accessed through the following symbolic names:

SMG$L_DEVCHAR	Device characteristics (longword).
SMG$L_DEVDEPEND	Specific characteristics 1 (longword).
SMG$L_DEVDEPEND2	Specific characteristics 2 (longword).
SMG$B_DEVCLASS	Device class (byte), for example, DC$_TERM.
SMG$B_SMG_DEVTYPE	Internal SMG device type (byte). The four possible values for SMG$B_SMG_DEVTYPE are as follows: SMG$K_UNKNOWN SMG$K_VTFOREIGN SMG$K_HARDCOPY SMG$K_VTTERMTABLE
SMG$B_PHY_DEVTYPE	Physical device type (byte), for example, TT$_VT100. The possible values for SMG$B_PHY_DEVTYPE are defined in $TTDEF in STARLET.
SMG$B_ROWS	Number of rows on pasteboard (byte).
SMG$W_WIDTH	Pasteboard width (word).

SMG$B_COLOR — Background color setting (byte). Valid values for SMG$B_COLOR are as follows:

SMG$C_COLOR_UNKNOWN	Unknown background color
SMG$C_COLOR_WHITE	Light background
SMG$C_COLOR_BLACK	Dark background
SMG$C_COLOR_BLUE	Blue background
SMG$C_COLOR_CYAN	Cyan (green-blue) background
SMG$C_COLOR_GREEN	Green background
SMG$C_COLOR_MAGENTA	Magenta background
SMG$C_COLOR_RED	Red background
SMG$C_COLOR_YELLOW	Yellow background
SMG$C_COLOR_LIGHT	White background
SMG$C_COLOR_DARK	Black background
SMG$C_COLOR_USER1	User-defined background 1
SMG$C_COLOR_USER2	User-defined background 2

SMG$B_PARITY — Parity attributes (byte). This field is zero if the pasteboard is not a terminal.

SMG$W_SPEED — Terminal speed (word). This field is zero if the pasteboard is not a terminal.

SMG$W_FILL — Fill characteristics (word). This field is zero if the pasteboard is not a terminal.

SMG$W_PHYS_CURSOR_ROW — Pasteboard row containing physical cursor (word).

SMG$W_PHYS_CURSOR_COL — Pasteboard column containing physical cursor (word).

SMG$L_CURSOR_DID — Display identifier of topmost display containing physical cursor (longword).

pasteboard-info-table-size

OpenVMS usage	longword_unsigned
type	longword (unsigned)
access	read only
mechanism	by reference

Specifies the number of bytes in the pasteboard information table. The **pasteboard-info-table-size** argument is the address of an unsigned longword that contains the size (in bytes) of the pasteboard information table.

The size you specify must be exact. You can specify this size with the symbolic constant SMG$C_PASTEBOARD_INFO_BLOCK.

Description

SMG$GET_PASTEBOARD_ATTRIBUTES gets pasteboard attributes and stores them in the pasteboard information table.

Return Values

SS$_NORMAL	Normal successful completion.
SMG$_INVARG	Incorrect size specified in **pasteboard-info-table-size.**
SMG$_WRONUMARG	Wrong number of arguments.

Example

```
10      !+
        !This VAX BASIC example demonstrates the use of the routine
        ! SMG$GET_PASTEBOARD_ATTRIBUTES.
        !-

        OPTION TYPE = EXPLICIT
        OPTION CONSTANT TYPE = INTEGER

        %INCLUDE "$SMGDEF" %FROM %LIBRARY "SYS$LIBRARY:BASIC$STARLET"
        %INCLUDE "$SSDEF"  %FROM %LIBRARY "SYS$LIBRARY:BASIC$STARLET"

        EXTERNAL LONG FUNCTION LIB$SIGNAL( LONG BY VALUE ),      &
                               SMG$CREATE_PASTEBOARD( LONG ),    &
                               SMG$GET_PASTEBOARD_ATTRIBUTES( LONG, ANY, LONG )

        DECLARE SMG$ATTRIBUTE_INFO_BLOCK SMG_INFO
        DECLARE LONG S, PASTEBOARD_ID

        S = SMG$CREATE_PASTEBOARD( PASTEBOARD_ID )
        IF S <> SS$_NORMAL THEN CALL LIB$SIGNAL( S ) END IF

        S = SMG$GET_PASTEBOARD_ATTRIBUTES( PASTEBOARD_ID, &
                                           SMG_INFO,      &
                                           SMG$C_PASTEBOARD_INFO_BLOCK )
        IF S <> SS$_NORMAL THEN CALL LIB$SIGNAL( S ) END IF

        PRINT SMG_INFO::SMG$L_DEV_CHAR          ! Device characteristics
        PRINT SMG_INFO::SMG$L_DEV_DEPEND        ! Specific characteristics (1)
        PRINT SMG_INFO::SMG$L_DEV_DEPEND2       ! Specific characteristics (2)
        PRINT SMG_INFO::SMG$B_DEV_CLASS         ! Device class ( DC$_TERM )
        PRINT SMG_INFO::SMG$B_PBD_TYPE          ! SMG type ( SMG$K_VTTERMTABLE )
        PRINT SMG_INFO::SMG$B_DEV_TYPE          ! Device type ( DT$_VT100 )
        PRINT SMG_INFO::SMG$B_NUM_ROWS          ! Number of rows on pasteboard
        PRINT SMG_INFO::SMG$W_NUM_COLUMNS       ! Number of cols on pasteboard
        PRINT SMG_INFO::SMG$B_PBD_COLOR         ! Pasteboard background color
        PRINT SMG_INFO::SMG$B_DEV_PARITY        ! Device parity characteristic
        PRINT SMG_INFO::SMG$W_DEV_SPEED         ! Device speed characteristic
        PRINT SMG_INFO::SMG$W_DEV_FILL          ! Device fill characteristic
        PRINT SMG_INFO::SMG$W_PHYS_CURSOR_ROW   ! Physical cursor row
        PRINT SMG_INFO::SMG$W_PHYS_CURSOR_COL   ! Physical cursor column
        PRINT SMG_INFO::SMG$L_DISPLAY_ID        ! Display_id containing cursor
        END
```

5.11 *OpenVMS RTL String Manipulation (STR$) Manual*

V6.1 The Bookreader version of this manual incorrectly lists the passing mechanism of the **start-position** argument in the STR$POSITION routine. The correct passing mechanism is by reference. This will be corrected in a future version of the Bookreader file.

5.12 System Management Documentation

V1.5 The appendix that lists and describes the files on a system disk, formerly located in the *OpenVMS System Manager's Manual,* has been moved into DCL Help.

To view this information, enter the following command:

```
$ HELP SYSTEM_FILES
```

To extract the list of system files to an output file in your directory, use the following command, where *file-name* represents the name of the output file:

```
$ HELP/OUTPUT=file-name  SYSTEM_FILES
```

5.13 *OpenVMS Utility Routines Manual*

V6.1 Two floating point data types that were developed with Version 1.5 to support the IEEE standard for binary floating-point arithmetic, ANSI/IEEE 754, were not included in the Version 1.5 documentation for the callable routines SOR$BEGIN_MERGE and SOR$BEGIN_SORT.

For Version Version T6.1-FT4, the two floating point data types have been added to the list of acceptable data types for the key_buffer argument for each of these routines:

- DSC$K_DTYPE_FS—supports IEEE single-precision S_floating point operations
- DSC$K_DTYPE_FT—supports IEEE double-precision T_floating point operations

The new data types specifically address the standard's definition of four floating-point formats in two groups, basic and extended, with each group having two widths, single and double.

For more detailed information, see the *Alpha Architecture Handbook.*

5.14 *VMScluster Systems for OpenVMS*

V1.5 Chapter 4 of the OpenVMS AXP Version 1.5 manual *VMScluster Systems for OpenVMS* states that you do not need to enable the DECnet extended function license DVNETRTG on an AXP node if one of the VMS Version 5.5–2 nodes is already a routing node. DVNETRTG is not the correct name of the license. The correct name of the extended function license for AXP nodes is DVNETEXT.

Index

How to Order Additional Documentation

Technical Support

If you need help deciding which documentation best meets your needs, call 800-DIGITAL (800-344-4825) and press 2 for technical assistance.

Electronic Orders

If you wish to place an order through your account at the Electronic Store, dial 800-234-1998, using a modem set to 2400- or 9600-baud. You must be using a VT terminal or terminal emulator set at 8 bits, no parity. If you need assistance using the Electronic Store, call 800-DIGITAL (800-344-4825) and ask for an Electronic Store specialist.

Telephone and Direct Mail Orders

From	Call	Write
U.S.A.	DECdirect Phone: 800-DIGITAL (800-344-4825) Fax: (603) 884-5597	Digital Equipment Corporation P.O. Box CS2008 Nashua, NH 03061
Puerto Rico	Phone: (809) 781-0505 Fax: (809) 749-8377	Digital Equipment Caribbean, Inc. 3 Digital Plaza, 1st Street Suite 200 Metro Office Park San Juan, Puerto Rico 00920
Canada	Phone: 800-267-6215 Fax: (613) 592-1946	Digital Equipment of Canada Ltd. 100 Herzberg Road Kanata, Ontario, Canada K2K 2A6 Attn: DECdirect Sales
International	————	Local Digital subsidiary or approved distributor
Internal Orders[1] (for software documentation)	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
Internal Orders (for hardware documentation)	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

[1] Call to request an Internal Software Order Form (EN–01740–07).

Reader's Comments

OpenVMS AXP Version T6.1–FT4
Release Notes
AA–Q1Z6B–TE

Your comments and suggestions help us improve the quality of our publications.

Thank you for your assistance.

I rate this manual's:	Excellent	Good	Fair	Poor
Accuracy (product works as manual says)	□	□	□	□
Completeness (enough information)	□	□	□	□
Clarity (easy to understand)	□	□	□	□
Organization (structure of subject matter)	□	□	□	□
Figures (useful)	□	□	□	□
Examples (useful)	□	□	□	□
Index (ability to find topic)	□	□	□	□
Page layout (easy to find information)	□	□	□	□

I would like to see more/less ______

What I like best about this manual is ______

What I like least about this manual is ______

I found the following errors in this manual:

Page Description

Additional comments or suggestions to improve this manual:

For software manuals, please indicate which version of the software you are using: ______

Name/Title ______ Dept. ______
Company ______ Date ______
Mailing Address ______
______ Phone ______