digital

# OpenVMS Compatibility Between VAX and AXP

OpenVMS

Part Number: AA–PYQ4A–TE

# OpenVMS Compatibility Between VAX and AXP

Order Number: AA–PYQ4A–TE

**May 1993**

This manual compares and contrasts OpenVMS on VAX and AXP computers, focusing on the features provided to end users, system managers, and programmers.

**Revision/Update Information:** This is a new manual.

**Software Version:** OpenVMS VAX Version 6.0

**Digital Equipment Corporation**
**Maynard, Massachusetts**

**May 1993**

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license and may be used or copied only in accordance with the terms of such license.

No responsibility is assumed for the use or reliability of software on equipment that is not supplied by Digital Equipment Corporation or its affiliated companies.

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

ZK6310

This document was prepared using VAX DOCUMENT, Version 2.1.

# Contents

## 3 Migration When You're Ready

## 4 Ensuring the Portability of Applications

## Index

## Examples

## Figures

## Tables

# Preface

## Intended Audience

This manual is of primary interest to current OpenVMS VAX users who are considering the addition of an OpenVMS AXP system to their computing environment and to users who are planning to migrate OpenVMS VAX applications to OpenVMS AXP systems.

## Document Structure

This manual consists of the following chapters and an index:

- Chapter 1 describes the similarities and differences between OpenVMS VAX and OpenVMS AXP and some of the new features introduced for both systems.
- Chapter 2 describes how OpenVMS VAX and OpenVMS AXP systems can interoperate in networks and VMSclusters.
- Chapter 3 lists the features available and planned for the OpenVMS AXP Version 1.5 operating system and for Digital layered products. It also lists some of the third-party applications that are available on OpenVMS AXP systems and describes the migration products and services available from Digital.
- Chapter 4 describes how to assess the portability of an OpenVMS VAX application. It also discusses the differences between the OpenVMS VAX and the OpenVMS AXP programming environments and presents guidelines for new program development on OpenVMS VAX.

## Associated Documents

To find out more about topics discussed in this manual, refer to the following table for the topic and related document.

| Topic | Document |
|---|---|
| Debugger features that contribute to migration | *OpenVMS Debugger Manual* |
| DECnet for OpenVMS AXP | *DECnet for OpenVMS Networking Manual* |
| DECnet for OpenVMS AXP utilities | *DECnet for OpenVMS Network Management Utilities* |
| Delta/XDelta changes for OpenVMS AXP | *OpenVMS Delta/XDelta Debugger Manual* |
| Device drivers, user-written, for OpenVMS AXP | *OpenVMS AXP Device Support: Creating a Step 1 Driver from an OpenVMS VAX Driver* |

| Topic | Document |
|---|---|
| DPML (Digital Portable Mathematics Library) | *DPML, Digital Portable Mathematics Library* |
| Help Message Utility | *OpenVMS System Messages: Companion Guide for Help Message Users* |
| Help Message Utility overview | *OpenVMS VAX Version 6.0 New Features Manual* |
| Linker changes for OpenVMS AXP | *OpenVMS Linker Utility Manual* |
| MACRO–64 assembler | *MACRO-64 Assembler for OpenVMS AXP Systems Reference Manual* |
| Managing OpenVMS on VAX and AXP computers | *A Comparison of System Management on OpenVMS AXP and OpenVMS VAX* |
| PALcode | *Alpha Architecture Reference Manual* |
| Planning for migration | *Migrating to an OpenVMS AXP System: Planning for Migration* |
| Porting applications written in mid- and high-level languages | *Migrating to an OpenVMS AXP System: Recompiling and Relinking* Applications |
| Porting VAX MACRO applications | *Migrating to an OpenVMS AXP System: Porting VAX MACRO Code* |
| SDA commands for OpenVMS AXP | *OpenVMS AXP System Dump Analyzer Utility Manual* |
| Translating OpenVMS VAX images into OpenVMS AXP images | *DECmigrate for OpenVMS AXP Systems Translating Images* |
| VAXcluster and VMScluster systems | *VMScluster Systems for OpenVMS* |
| VMScluster system restrictions | *OpenVMS AXP Version 1.5 Release Notes* |

## 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 used in this manual:

**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.

-
A hyphen in code examples indicates that additional arguments to the request are provided on the line that follows.

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

# 1 Overview

This chapter describes:

- Lineage of OpenVMS AXP
- Similarities and differences between OpenVMS VAX and OpenVMS AXP in the end-user, system management, and programming environments

## 1.1 Introduction

The purpose of this manual is to provide OpenVMS VAX users with the information they need to assess the impact of adding one or more OpenVMS AXP Version 1.5 systems to their computing environments. The manual focuses on the similarities and differences between OpenVMS AXP Version 1.5 and VMS Version 5.4–2, the operating system on which it is based. Where applicable, comparisons are also made with later versions of the OpenVMS operating system.

OpenVMS AXP runs on AXP computers, Digital's line of reduced instruction set computers (RISC). OpenVMS AXP systems provide as much compatibility as possible with OpenVMS VAX systems without compromising the advantages offered by the Alpha AXP architecture. OpenVMS VAX customers can use the RISC technology of OpenVMS AXP with little change in their computing environment.

Experienced OpenVMS VAX system managers will find their knowledge and most of their practices transferable to the OpenVMS AXP environment. Many OpenVMS AXP system managers compare the degree of change to that introduced by VAX VMS Version 5.0.

Most of Digital's customers use OpenVMS VAX to run their organization's critical applications. They depend on its reliability, robust engineering, and leadership features, such as VAXcluster systems. Digital anticipates that many of its customers, as their organizations expand, will add OpenVMS AXP systems to their computing environments. Digital recognizes that this is an evolutionary process and will differ from customer to customer. Digital is committed to providing a clear coexistence environment and migration path between OpenVMS VAX and OpenVMS AXP.

## 1.2 Lineage of OpenVMS AXP

Ever since Version 1.0 of the VMS operating system was released for VAX computers in 1978, it has evolved to include new features and new technologies. Each new release represents a balance between compatibility with earlier releases and the introduction of new features and new technology that enable users to do their work more cost-effectively. A recent development in this evolution is the introduction of OpenVMS AXP in November 1992.

VMS Version 5.4–2 is the version on which OpenVMS AXP Version 1.0 is based. There is a high level of compatibility between the two systems, although some differences exist. For example, OpenVMS AXP contains come features introduced on OpenVMS VAX after VMS Version 5.4–2, such as DECthreads and the parallel processing run-time library routines (PPL RTL). Some OpenVMS VAX features are not yet available on OpenVMS AXP systems, such as Volume Shadowing. Table 3–1 shows the availability of selected base operating system features.

## 1.3 End-User's Environment

The end-user's environment on OpenVMS AXP Version 1.5 is virtually the same as that on VMS Version 5.4–2, except for the addition of some new features developed after the release of VMS Version 5.4–2.

### DIGITAL Command Language (DCL)

The DIGITAL Command Language (DCL), the standard user interface to OpenVMS, remains essentially unchanged with OpenVMS AXP. All commands, qualifiers, and lexical functions available on VMS Version 5.4–2 also work on OpenVMS AXP.

### DCL Help

DCL help is available on OpenVMS AXP Version 1.5. Most of the DCL help text is common to both OpenVMS AXP and OpenVMS VAX systems. For a few topics, some information that is specific to OpenVMS VAX and some that is specific to OpenVMS AXP is included in the display for both systems. This system-specific information is identified by the names VAX and AXP.

### Command Procedures

Command procedures that use commands, qualifiers, and lexical functions available on VMS Version 5.4–2 or earlier versions will continue to work on an OpenVMS AXP system without change.

### DECforms

The DECforms interface is unchanged.

### Editors

The EVE and EDT editors are unchanged. However, EVE is the default editor for OpenVMS AXP Version 1.5 and for OpenVMS VAX Version 6.0. EDT was the default editor for VMS Version 5.4–2 and subsequent versions of OpenVMS VAX.

The DSR and TECO editors are also provided with OpenVMS AXP. They, too, are unchanged.

### Databases

Standard Digital databases (such as Rdb/VMS) function the same on OpenVMS VAX and OpenVMS AXP systems.

### DECwindows Motif

DECwindows Motif on OpenVMS AXP is unchanged from DECwindows Motif on OpenVMS VAX except for the addition of support for a font server and scalable fonts on OpenVMS AXP systems. For more information about the new font features, see the *OpenVMS AXP Version 1.5 Release Notes*.

### Help Message Utility

The Help Message utility, a new feature for OpenVMS VAX Version 6.0, is available on OpenVMS AXP Version 1.5. It enables users to access online descriptions of system messages on a character-cell terminal (including DECterm windows).

Help Message operates through a DCL interface that accesses message descriptions in a text file. This text file is derived from the latest version of the OpenVMS system messages documentation and, optionally, from other source files, including user-supplied message documentation. For more information about Help Message, see the *OpenVMS VAX Version 6.0 New Features Manual* and *OpenVMS System Messages: Companion Guide for Help Message Users*.

### Password Generator

Both OpenVMS AXP Version 1.5 and VMS Version 5.4–2 have random password generators. However, the random password generator on OpenVMS AXP systems has a new mechanism for generating passwords. This new mechanism produces nonsense passwords such as *dramnock*, *inchworn*, *mycousia*, and *pestrals* that seem "natural" to users. It makes possible the future development of language-specific random passwords. An outgrowth of this new mechanism is that the list of possible passwords presented by an OpenVMS AXP system is not hyphenated.

# 1.4 System Manager's Environment

The similarities and differences in the system manager's environment between VMS Version 5.4–2 and OpenVMS AXP Version 1.5 are described in this section. The differences include the addition of some new features developed after the release of VMS Version 5.4–2.

## 1.4.1 Similarities

Most of the VMS Version 5.4–2 system management utilities, command formats, and tasks are identical in the OpenVMS AXP environment.

The security features of VMS Version 5.4–2 are the same on OpenVMS AXP Version 1.5. They do not include the C2 security features of OpenVMS VAX Version 6.0. A number of system resources, rights database characteristics, and DCL commands are affected by C2 features in OpenVMS VAX Version 6.0. If you are migrating to OpenVMS AXP Version 1.5 from a version of OpenVMS VAX earlier than Version 6.0, you will not notice any security-related changes.

## 1.4.2 Differences

There are some differences that must be considered to properly set up, maintain, secure, and optimize OpenVMS AXP systems and to establish proper network connections. These differences are briefly described in this section. For more detailed information on these differences, see *A Comparison of System Management on OpenVMS AXP and OpenVMS VAX.*

### 1.4.2.1 Different Page Size

OpenVMS VAX and OpenVMS AXP systems allocate and deallocate memory for processes in units called **pages**. A page on an OpenVMS VAX system is 512 bytes. On OpenVMS AXP systems, the page size will be one of four values, 8K, 16K, 32K, or 64K bytes. An OpenVMS AXP system implements only one of the four page sizes; the initial set of AXP computers use an 8K byte (8192 bytes) page.

This difference in page size is significant to OpenVMS system managers in two ways:

- You might need to adjust process quotas and limits, and system parameters, to account for the additional resources (especially memory resources) users might require. For example, higher values for the PGFLQUOTA process quota and the GBLPAGES system parameter might be necessary.
- In a number of cases, OpenVMS AXP interactive utilities present to and accept from users units of memory in a 512-byte quantity called a **pagelet**. Thus, one AXP pagelet is the same size as one VAX page. On an AXP computer with 8K byte pages, 16 AXP pagelets equal 1 AXP page.

In your OpenVMS AXP environment, you will need to notice when page or pagelet values are being shown in memory displays. If a memory value represents a page on an AXP system, the documentation might refer to "CPU-specific pages." This convention indicates possible significant differences in the size of the memory being represented by the page unit, depending on the AXP computer in use (8K, 16K, 32K, or 64K byte pages). In general, OpenVMS AXP utilities display CPU-specific page values when the data represents physical memory.

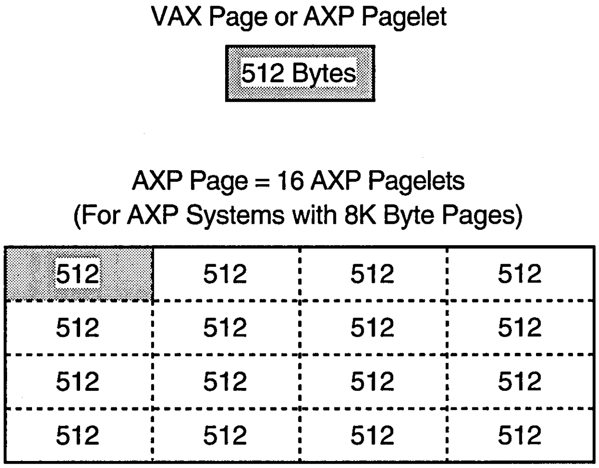
Internally, for the purposes of memory allocation, deletion, and protection, OpenVMS AXP will round up (if necessary) the value you supply in pagelets to a number of CPU-specific pages.

The use of pagelets provides compatibility for OpenVMS VAX users, system managers, and application programmers who are accustomed to thinking about memory values in 512-byte units. In a VMScluster system with OpenVMS VAX Version V5.5–2 nodes and OpenVMS AXP Version 1.5 nodes, it is helpful to know that a VAX page and an AXP pagelet represent a common unit of 512 bytes. Also, existing OpenVMS VAX applications do not need to change parameters to the memory management system services when the applications are ported to OpenVMS AXP.

OpenVMS AXP does not allocate or deallocate a portion of a page. The user-interface quantity called a pagelet is not used internally by the operating system. Pagelets are accepted and displayed by utilities so that users and applications operate with the information that each VAX page value and each AXP pagelet value equal a common 512-byte quantity.

Figure 1–1 illustrates the relative sizes of a VAX page, an AXP 8K byte page, and an AXP pagelet.

**Figure 1–1 Comparison of VAX and AXP Page Size**



ZK-6059A-GE

#### 1.4.2.2 VMScluster Systems Supported

A **VMScluster** system, known as a VAXcluster system in VAX environments, can consist of either of the following:

- All OpenVMS AXP Version 1.5 nodes
- A combination of OpenVMS AXP Version 1.5 nodes and OpenVMS VAX Version V5.5–2 nodes

For more information on VMScluster systems, see Section 2.3 in this manual, the *OpenVMS AXP Version 1.5 Release Notes*, and *VMScluster Systems for OpenVMS*.

#### 1.4.2.3 User-Written Device Drivers

OpenVMS AXP Version 1.5 includes a mechanism to allow essential OpenVMS VAX drivers to be ported to OpenVMS AXP systems quickly and with minimal changes. This mechanism, known as the Step 1 driver interface, is meant to be only a temporary way of providing device support for OpenVMS AXP systems.

Although a manual and training are available for those with critical needs, this approach is not recommended. A future OpenVMS AXP release will provide official support for Step 2 OpenVMS AXP drivers. Digital is designing the Step 2 interface to allow drivers to be written in C and conform to the OpenVMS calling standard.

#### 1.4.2.4 I/O Subsystem Configuration Commands

On OpenVMS VAX systems, the System Generation utility (SYSGEN) is used to configure the I/O subsystem, as shown in Table 1–1. On OpenVMS AXP systems, SYSGEN is used for some configuration tasks, and the System Management utility (SYSMAN) and the AUTOGEN command procedure are used for others, as shown in Table 1–1.

Table 1–1 A Comparison of I/O Subsystem Configuration

| | OpenVMS VAX | OpenVMS AXP |
|---|---|---|
| SYSGEN | Modify system parameters[1]<br>Load page and swap files<br>Create additional page files<br>Create additional swap files<br>Load device drivers | Modify system parameters[1]<br>Load page and swap files<br>Create additional page files<br>Create additional swap files |
| SYSMAN | Not used for these functions | Load device drivers |

[1] Although SYSGEN is available for modifying system parameters, Digital recommends that you use AUTOGEN and its data files instead, or that you use SYSMAN between boots, for dynamic parameters.

OpenVMS VAX command procedures that use commands such as SYSGEN AUTOCONFIGURE ALL must be modified if they are copied to OpenVMS AXP systems as part of your migration effort.

#### 1.4.2.5 MONITOR POOL Command

The DCL command MONITOR POOL that is used on OpenVMS VAX Version 5.5 and earlier systems is not provided on OpenVMS AXP systems or on OpenVMS VAX Version 6.0 systems. MONITOR POOL functions are replaced by enhanced, adaptive pool management functions and two System Dump Analyzer (SDA) commands in OpenVMS AXP: SHOW POOL/RING_BUFFER and SHOW POOL /STATISTICS.

#### 1.4.2.6 Higher Disk Quotas

You might need to increase disk quotas on OpenVMS AXP disks that store translated OpenVMS VAX images and native OpenVMS AXP images. Translated images are OpenVMS AXP executable images produced by the VAX Environment Software Translator (VEST) component of DECmigrate (described in Section 3.8). Translated images require more disk space because each image includes both AXP code and the original VAX code. Native OpenVMS AXP images require more disk space because RISC images typically contain more instructions and code to establish the linkage between procedure calls. The default values for related memory quotas have been adjusted on OpenVMS AXP systems.

#### 1.4.2.7 OpenVMS File Names

The names of some command procedure files supplied by the operating system have changed. For example, SYSTARTUP_V5.COM from OpenVMS VAX Version 5.5 and earlier versions is called SYSTARTUP_VMS.COM on OpenVMS AXP as it is on OpenVMS VAX Version 6.0. The VAXVMSSYS.PAR system parameter file is called ALPHAVMSSYS.PAR on OpenVMS AXP systems. On OpenVMS VAX systems, the VAXVMSSYS.PAR system parameter file name remains the same.

#### 1.4.2.8 Some Operating System Functions Not Planned

The Patch utility and the card reader driver and input symbiont are not planned for OpenVMS AXP systems.

#### 1.4.2.9 Some Layered Products Not Supported

For OpenVMS AXP Version 1.5, some Digital layered products are not supported. If you copy existing startup procedures from one of your OpenVMS VAX computers to an OpenVMS AXP computer, you must comment out the calls to the startup procedures of currently unsupported layered products.

The availability of many layered products is shown in Table 3–2. In the United States and Canada, you can obtain a catalog that lists all the available Digital layered products and third-party applications by calling 1-800-DEC-INFO (1-800-332-4636). In other locations, you can obtain the catalog from your Digital account representative or authorized reseller.

## 1.5 Programming Environment

The similarities and differences in the programmer's environment between VMS Version 5.4–2 and OpenVMS AXP Version 1.5 are outlined in this section and described in more detail in Chapter 4, except where noted otherwise.

### 1.5.1 Similarities

Most DEC compilers (and the MACRO–64 assembler) are available on OpenVMS AXP Version 1.5 as shown in the following list:

- DEC Ada
- DEC C
- DEC COBOL
- DEC Fortran
- DEC Pascal
- MACRO–32
- OPS5

For the release dates of compilers not yet available, see Table 3–2.

Except for the MACRO–32 compiler, these compilers are also available on OpenVMS VAX Version 6.0. (A MACRO-32 cross-compiler and cross-compilers for DEC Fortran, DEC C, and Bliss are available on OpenVMS VAX systems (see Section 3.6.1).)

The same types of program development tools that programmers are accustomed to using on OpenVMS VAX are available on OpenVMS AXP systems including the Linker utility, the Librarian utility, the Delta/XDelta Debugger, the OpenVMS Debugger (also known as the symbolic debugger), and run-time libraries, including the PPL RTL.

### 1.5.2 Differences

Before moving any OpenVMS VAX applications to an OpenVMS AXP system, Digital recommends that you become familiar with the following differences in the OpenVMS AXP program development environment:

- **DEC Compilers**
  - Many new Digital compilers have been released on both AXP and VAX computers. The names of these compilers begin with DEC, such as DEC Fortran and DEC C. They use a common back-end which optimizes the performance of the code on both VAX and AXP computers. The names of the earlier compilers begin with VAX and are not available on AXP

computers. The differences between the VAX compilers and the DEC compilers and the differences between the DEC compilers' operation on OpenVMS VAX and their operation on OpenVMS AXP may require some minor changes to your code.

The MACRO–32 compiler, running on OpenVMS AXP, converts VAX MACRO code into AXP executable code. It is provided to ease migration.

– The RISC architecture of Alpha AXP differs from the complex instruction set computer (CISC) architecture of the VAX. The differences are particularly apparent when multiple processes or multiple execution threads in the same process access the same region of memory. An asynchronous system trap (AST) routine whose execution preempts the processing of a main routine is one example of concurrent threads within a single process.

  The VAX architecture, through its microcode, provides instruction semantic guarantees that the Alpha AXP architecture does not. Complex atomic operations and synchronization guarantees incur overhead that RISC architectures are designed to avoid.

  For example, on a VAX computer, you can perform complex memory operations atomically and can access data at byte and word memory locations. If your code contains such VAX architectural dependencies, you likely will need to either use a compiler qualifier that mitigates the dependencies or make changes to your code.

– Although most of the Digital compilers are available, a few are not available yet. If the native compiler is not available, you can translate user-mode code with DECmigrate for OpenVMS AXP (see Section 3.8).

– Support for H-floating and full-precision D-floating has been eliminated from hardware to improve overall system performance.

  AXP hardware converts D-floating data to G-floating for processing. On VAX systems, D-floating has 56 fraction bits (D56) and 16 decimal digits of precision.

  The H-floating and D-floating data types can usually be replaced by G-floating or one of the IEEE formats. However, if you require H-floating or the extra precision of D56 (56-bit D-floating), you may have to translate part of your application.

- **Native assembler**

  A native assembler is not bundled with the OpenVMS AXP operating system. The native assembler, MACRO–64, is available as a layered product. However, the MACRO–32 compiler for OpenVMS AXP is bundled with OpenVMS AXP. You can use it to compile your programs written in VAX MACRO into AXP machine code, although some modifications may be required. Digital recommends that you write any new code for AXP computers in mid- and high-level languages whenever possible.

- **Cross-compilers**

  Digital offers an AXP Migrations Tools Package with cross-compilers that you can run on OpenVMS VAX systems to produce applications that run on OpenVMS AXP. For more information about this package, see Section 3.6.1.

- **DECmigrate for OpenVMS AXP**

  DECmigrate for OpenVMS AXP is a Digital layered product that translates OpenVMS VAX images into OpenVMS AXP images. It also can be used to analyze code to determine how easy or difficult it would be to migrate it. For more information about DECmigrate for OpenVMS AXP, see Section 3.8.

- **Linker**

  The way certain linking tasks, such as creating shareable images, are performed is different on OpenVMS AXP systems. You may need to modify the LINK command used to build your application. For example, instead of creating a transfer vector file for a shareable image, you must create a linker options file and declare universal symbols by specifying the SYMBOL_VECTOR= option.

- **Librarian**

  The Librarian utility provides a new qualifier, /VAX. The /VAX qualifier directs the Librarian utility to create an OpenVMS VAX object module library when used with the /CREATE and /OBJECT qualifiers or an OpenVMS shareable image library when used with the /SHARE qualifier. OpenVMS AXP libraries are the default on OpenVMS AXP systems.

- **OpenVMS Debugger**

  The OpenVMS Debugger provides several features specific to OpenVMS AXP that facilitate debugging OpenVMS AXP code. These features address the architectural differences that exist between VAX and AXP computers. For example, the /UNALIGNED_DATA qualifier used with the SET command enables you to detect unaligned data.

- **Delta/XDelta Debugger**

  The Delta/XDelta Debugger provides several new commands and changes to existing commands for debugging OpenVMS AXP programs.

- **System Dump Analyzer (SDA)**

  Because of the architectural differences, a new system dump analyzer, the OpenVMS AXP System Dump Analyzer utility, was created. Some of the commands of the OpenVMS VAX System Dump Analyzer utility were modified or extended.

- **No vector processing on AXP**

  Vector processors were an option for VAX 6500 and VAX 9000 computers to provide higher performance for some numerically intensive applications. AXP computers and later versions of VAX computers do not provide this option because their basic designs provide high-speed calculations. If you used this option for Fortran applications, you do not have to make any changes to your code for it to run on later models of VAX or AXP computers. You only have to recompile it with the DEC Fortran for OpenVMS AXP compiler. If you used vector-specific support in VAX MACRO applications, you will need to make changes to your code before recompiling it with the MACRO–32 compiler for OpenVMS AXP. Note that you cannot translate image files whose source files included vector instructions. The VAX Environment Software Translator (VEST) component of DECmigrate does not support them.

For information on ensuring the portability of your OpenVMS VAX applications, see Chapter 4.

# 2 Interoperability of OpenVMS VAX and OpenVMS AXP

This chapter describes the similarities and differences of the following topics:

- Interoperability of OpenVMS VAX and OpenVMS AXP on a DECnet network and on a TCP/IP network
- DECnet network features and management
- Interoperability of OpenVMS VAX and OpenVMS AXP in a VMScluster

DECnet for OpenVMS AXP Version 1.5 is used to establish networking connections with other OpenVMS VAX and OpenVMS AXP nodes. DECnet for OpenVMS, previously known as DECnet–VAX, implements Phase IV of DNA (DIGITAL Network Architecture). The features of DECnet for OpenVMS AXP are similar to those of the DECnet–VAX software that is part of VMS Version 5.4–2, with a few exceptions.

TCP/IP is the network enabled by DEC TCP/IP Services for OpenVMS. It provides interoperability and resource sharing between OpenVMS systems, UNIX systems, and other systems that support the TCP/IP and NFS protocol suites. The resource sharing includes the services that users of DECnet are familiar with: remote file access, remote terminal access, remote command execution, remote printing, mail, and application development.

A VMScluster system is very similar to a VAXcluster system, except that it can consist entirely of OpenVMS AXP nodes or as a combination of OpenVMS VAX and OpenVMS AXP nodes. The system management of a VMScluster system is essentially the same.

VMScluster systems for OpenVMS AXP Version 1.5 offer virtually all of the software features of VAXcluster systems with some configuration restrictions. Some of the related VAXcluster software products, such as Volume Shadowing, are not yet available for OpenVMS AXP nodes.

## 2.1 Interoperability on a Network

OpenVMS AXP and OpenVMS VAX systems can interoperate in a network, sharing system resources and enabling communication with local and remote nodes, in the same way that OpenVMS VAX systems interoperate. Network management of OpenVMS AXP nodes is also similar to network management of OpenVMS VAX nodes. However, some differences exist because not all features are supported on OpenVMS AXP systems at this time. The similarities and differences of network interoperability and management are discussed in this section.

## 2.1.1 Network Interfaces

Most of the network protocols, buses, and interconnects that are supported on OpenVMS VAX systems are also supported on OpenVMS AXP systems. For some network interfaces, support is either not planned or not yet available, as shown in the following tables.

### 2.1.1.1 Network Protocols

The network protocols supported on OpenVMS VAX and OpenVMS AXP systems are shown in Table 2–1. In some cases, the layered products that use these protocols are not yet available, as shown in Table 2–4.

**Table 2–1 Supported Network Protocols**

| Protocol | OpenVMS VAX V5.5-2 | OpenVMS VAX V6.0 | OpenVMS AXP V1.5 |
|---|---|---|---|
| DECnet (Phase IV) | Yes | Yes | Yes |
| DECnet/OSI | Yes | Yes | No[1] |
| LAD/LAST | Yes | Yes | Yes |
| LAT | Yes | Yes | Yes |
| LAVC | Yes | Yes | Yes |
| TCP/IP[2] | Yes | Yes | No[3] |
| X.25 | Yes | No | No[4] |

[1] Planned for a future release.

[2] Provided by DEC TCP/IP Services for OpenVMS (formerly named the VMS/Ultrix Connection).

[3] Planned for a release in the third quarter of 1993.

[4] Although VAX P.S.I. is not supported, an OpenVMS AXP node can connect to an X.25 network via an X.25 router on the same local area network via DEC X.25 Client for OpenVMS AXP.

### 2.1.1.2 Buses

The buses supported on OpenVMS VAX and OpenVMS AXP systems are shown in Table 2–2. Support is also dependent on the computer model. The buses supported by each operating system are not available for all computer models.

**Table 2–2 Bus Support**

| Bus | OpenVMS VAX V5.5-2 | OpenVMS VAX V6.0 | OpenVMS AXP V1.5 |
|---|---|---|---|
| BI-bus | Yes | Yes | No[1] |
| DSSI | Yes | Yes | Yes |
| EISA bus | No | No | No[2] |
| Futurebus+ | No | No | No[3] |
| Q–bus | Yes | Yes | No[1] |
| SCSI | Yes | Yes | Yes |
| TURBOchannel | Yes | Yes | Yes |
| UNIBUS | Yes | Yes | No[1] |
| VME | Yes | Yes | No[3] |
| XMI | Yes | Yes | Yes |

[1] Not planned for a future release.

[2] Planned for a future release (not supported on OpenVMS VAX)

[3] Planned for a future release.

### 2.1.1.3 Interconnects

The interconnects (including their respective protocols and drivers) that are supported on OpenVMS VAX and OpenVMS AXP systems are shown in Table 2–3.

**Table 2–3 Interconnect Support**

| Interconnect | DECnet[1] | TCP/IP[2] | LAT | Cluster: Computer to Computer[3] | Computer to Tape or Disk |
|---|---|---|---|---|---|
| Asynch Line | V | – | – | – | – |
| Synch Line | V | – | – | – | – |
| CI | V | – | – | A,V | A,V |
| DSSI | – | – | – | A,V | A,V |
| Ethernet | A,V | A,V | A,V | A,V | – |
| FDDI | A,V | A,V | A,V | V[4] | – |
| SCSI | – | – | – | – | A,V |

[1] Provided by DECnet for OpenVMS.

[2] Provided by DEC TCP/IP Services for OpenVMS.

[3] OpenVMS VAX systems can serve TMSCP compliant tapes to all nodes in a VMScluster system; OpenVMS AXP systems cannot at this time.

[4] An OpenVMS AXP system does not use an FDDI adapter for cluster communication at this time, but Ethernet bridging to FDDI backbones can be used.

**Key**

A = OpenVMS AXP Version 1.5
V = OpenVMS VAX Version V5.5-2 and Version 6.0
– = Neither

### 2.1.2 DECnet Network File Transfers

File transfers over the DECnet network, including copying and printing, can be done between OpenVMS VAX and OpenVMS AXP systems.

### 2.1.3 DECnet Remote Login

There is full interoperability for DECnet remote login between OpenVMS VAX and OpenVMS AXP nodes running DECnet (via the SET HOST command).

### 2.1.4 TCP/IP Network File Transfers

File transfers over a TCP/IP network are the same between OpenVMS AXP systems and UNIX systems as they are between OpenVMS VAX systems and UNIX systems.

### 2.1.5 TCP/IP Remote Login

Remote login over a TCP/IP network is the same between OpenVMS AXP systems and UNIX systems as it is between OpenVMS VAX systems and UNIX systems.

## 2.2 DECnet Network Features and Management

The similarities and differences in the network features and management for DECnet for OpenVMS AXP Version 1.5 and for DECnet–VAX for VMS Version 5.4–2 are described in this section. DECnet/OSI for OpenVMS is also discussed.

### 2.2.1 Similarities

The features and management of DECnet for OpenVMS AXP Version 1.5 are similar to that of DECnet–VAX for VMS Version 5.4–2, with some exceptions. The following list shows the features and management tasks that are identical:

- DECnet objects
- DECnet Test Sender/DECnet Test Receiver utility (DTS/DTR)
- Downline load and upline dump operations
- Event logging
- Ethernet monitor (NICONFIG)
- File access listener (FAL)

  (Identical between AXP nodes and between AXP and VAX nodes.)
- Loopback mirror testing
- NETCONFIG_UPDATE.COM procedure
- Node name rules
- Product Authorization Key (PAK) name for end-node license (DVNETEND)
- SET HOST capabilities

  (Identical between OpenVMS AXP nodes and between OpenVMS AXP and OpenVMS VAX nodes.)

- Size of network

  (The Phase IV limit on maximum size is 1023 nodes per area and 63 areas in the entire network. When DECnet/OSI for OpenVMS (also known as Phase V) is installed on OpenVMS VAX Version 5.5 systems, the maximum network size is much larger.)

- Task-to-task communication

### 2.2.2 Differences

The differences in the DECnet features and management between OpenVMS AXP and VMS Version 5.4-2 are shown in Table 2–4. Many of the differences will be eliminated in future releases of OpenVMS AXP and DECnet for OpenVMS AXP.

**Table 2–4 Differences of DECnet Features and Management Tasks**

| Feature or Task | OpenVMS VAX | OpenVMS AXP |
|---|---|---|
| Cluster alias | Level 1 and Level 2 routing supported on nodes acting as routers for a cluster alias. | Only Level 1 routing supported on nodes acting as routers for a cluster alias. |
| Configuring DECnet databases and starting OpenVMS AXP computer's access to the network | | Process for both is the same but subject to the routing limitations and the lack of support for CI, DDCMP, the Distributed Name Service (DNS) node name interface, VAX P.S.I., certain Network Control Program (NCP) utility command parameters, and DECnet/OSI. The functions of the SYS$MANAGER:STARTNET.COM procedure are similar. |
| DECnet/OSI for OpenVMS | Supported on OpenVMS VAX Version 5.5 and later versions, but DECnet/OSI for OpenVMS not yet available for OpenVMS VAX Version 6.0. | Not supported at this time; planned for a future release. |
| Distributed Name Service (DNS) node name interface | Supported on OpenVMS VAX Version 5.5 and later versions. | Not supported at this time; planned for a future release. |
| Lines supported | CI, asynch (DDCMP), Ethernet, and FDDI | Ethernet and FDDI |
| NETCONFIG.COM procedure (part for specifying a router) | NETCONFIG.COM prompts you, "Do you want to operate as a router?" | NETCONFIG.COM does not prompt you. You have to manually enable Level 1 routing. Otherwise, NETCONFIG.COM is the same on OpenVMS AXP systems. |
| Network management via Network Control Program (NCP) utility and the network management listener (NML) object | | In many cases, the NCP commands and parameters are identical. However, the NCP command parameters for SET and SHOW operations for DDCMP, full host-based routing, the Distributed Name Service (DNS), and VAX P.S.I. have no effect on OpenVMS AXP. |

(continued on next page)

Table 2–4 (Cont.) Differences of DECnet Features and Management Tasks

| Feature or Task | OpenVMS VAX | OpenVMS AXP |
|---|---|---|
| Product Authorization Key (PAK) name for cluster alias routing support | DVNETRTG | DVNETEXT |
| Routing | Level 1 and Level 2 routing are supported. | Level 1 routing is supported only on nodes acting as routers for a cluster alias. Level 2 routing is not supported. |
| VAX P.S.I. | Supported | Not supported. For DECnet for OpenVMS AXP Version 1.5, DEC X.25 Client for OpenVMS AXP replaces VAX P.S.I. |

For more information about the differences between DECnet for OpenVMS on VAX and AXP nodes, see *A Comparison of System Management on OpenVMS AXP and OpenVMS VAX*. For more information about DECnet for OpenVMS AXP Version 1.5, see *DECnet for OpenVMS Networking Manual* and *DECnet for OpenVMS Network Management Utilities.*

### 2.2.3 DECnet/OSI for OpenVMS

DECnet/OSI for OpenVMS is an OSI-compliant product. Although available for VAX computers, it is not yet available for AXP computers. DECnet/OSI for OpenVMS conforms to networking standards defined by the International Organization for Standardization (ISO). DECnet/OSI for OpenVMS features include:

- Larger networks than the 64,449-node limit in Phase IV networks
- A longer, more descriptive format for node names
- Use of a local name database or a distributed namespace database

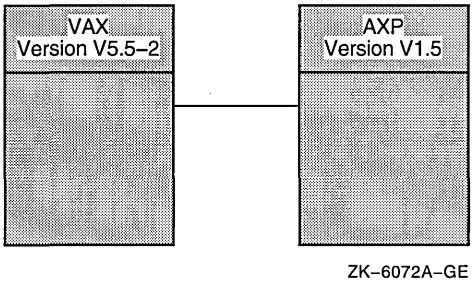
## 2.3 Interoperability in a VMScluster System

A VMScluster system, introduced with OpenVMS AXP Version 1.5, consists of the following:

- Multiple AXP computers running OpenVMS AXP Version 1.5 with the VMScluster software
- One or more AXP systems running OpenVMS AXP Version 1.5 with one or more VAX systems running OpenVMS VAX Version V5.5–2 with VAXcluster software

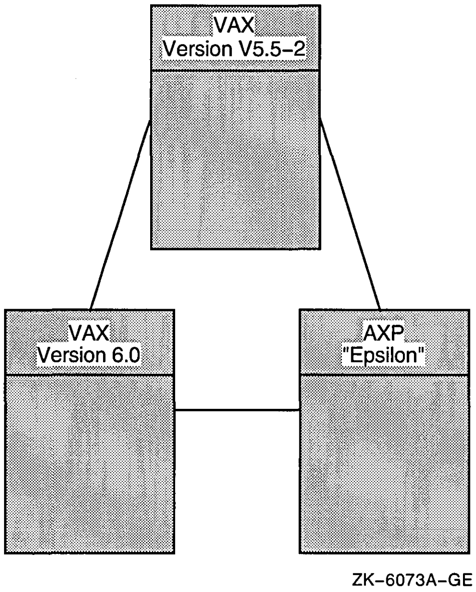
Figure 2–1 shows the coexistence of VAX computers running OpenVMS VAX Version V5.5–2 and AXP computers running OpenVMS AXP Version 1.5 in a VMScluster. For configurations with VAX and AXP nodes, each architecture needs its own system disk for booting.

**Figure 2–1 VMScluster Coexistence for OpenVMS AXP Version 1.5**



For OpenVMS AXP Version “Epsilon,” Digital plans to provide VMScluster coexistence for both OpenVMS VAX Version V5.5–2 and Version 6.0, as shown in Figure 2–2.

**Figure 2–2 VMScluster Coexistence for OpenVMS AXP Version “Epsilon”**



Not all software products for VAXclusters systems are available for VMScluster systems at this time. For a timetable of the availability of Digital layered products, see Table 3–2.

## 2.3.1 VMScluster Features

All of the VAXcluster features are available on VMScluster systems for OpenVMS AXP Version 1.5, except for tape serving by AXP nodes. These features include:

- Shared file system—all systems share read and write access to disk files in a fully coordinated environment. (However, in a VMScluster system with VAX and AXP nodes, each architecture needs its own system disk for booting.)
- Shared batch and print queues accessible from any system in the VMScluster system.
- OpenVMS lock manager system services operate for all nodes in a VMScluster system.

- All physical disks in a VMScluster system can be made accessible to all systems. (However, in a VMScluster system with both VAX and AXP nodes, each architecture needs its own system disk for booting.)
- All TMSCP compliant tapes can be made accessible to all systems. OpenVMS VAX nodes can serve TMSCP compliant tapes to all systems in a VMScluster system; OpenVMS AXP systems cannot.
- Process information and control services are available to application programs and system utilities on all nodes in a VMScluster system.
- Configuration command procedures assist in adding and removing systems and in modifying their configuration characteristics.
- The Show Cluster utility displays the status of all VMScluster hardware components and communication links.
- Standard OpenVMS system and security features work in a VMScluster system such that the entire VMScluster system operates as a single security domain.
- The VMScluster software balances the interconnect I/O load in VMScluster configurations that include multiple interconnects.
- Multiple VMScluster systems can be configured on a single or extended local area network (LAN).
- The /CLUSTER qualifier to the Monitor utility operates across the entire VMScluster.

### 2.3.2 VMScluster Configuration Support

VMScluster Software for OpenVMS AXP Version 1.5 supports the following interconnects for AXP computers:

- Computer Interconnect (CI)
- Ethernet
- Digital Storage System Interconnect (DSSI)

  DSSI is supported on DEC 4*nnn* systems but not on DEC 7*nnn* systems at this time.
- FDDI as a backbone (no VMScluster support for AXP computers)

VMScluster software support for FDDI adapters on AXP computers is planned for future releases.

A maximum of 96 systems can be configured in a VMScluster system, the same limit as in a VAXcluster system. However, there are limits to the number of OpenVMS AXP systems in a VMScluster. For configurations supported in a VMScluster system, see the Software Product Description (SPD) for VMScluster Software for OpenVMS AXP, Version 1.5.

### 2.3.3 VMScluster Software Capabilities Not Supported

The following software capabilities are not supported at this time for VMScluster systems for OpenVMS AXP Version 1.5:

- Volume shadowing on AXP systems
- Disk striping on AXP systems
- Accessing VAX based shadow sets from AXP systems

- Accessing VAX based stripe sets from AXP systems
- AXP systems serving tape drives to AXP systems or VAX systems
- VAX satellites booting from AXP systems
- AXP satellites booting from VAX systems
- VAX and AXP systems sharing a single system disk

For more information about configuration rules and recommendations, refer to the SPD for VMScluster Software for OpenVMS AXP, Version 1.5. For information about the differences between managing VAXclusters and VMSclusters, see *A Comparison of System Management on OpenVMS AXP and OpenVMS VAX*. For all other VMScluster information, refer to *VMScluster Systems for OpenVMS* and to the *OpenVMS AXP Version 1.5 Release Notes*.

# 3 Migration When You're Ready

This chapter provides information that will help you decide when to include one or more OpenVMS AXP systems in your computing environment and which migration resources to use. The following topics are discussed:

- Availability of base operating system features
- Timetable for the availability of Digital layered products
- Available third-party applications
- Application migration paths
- Hardware and software investment protection programs
- Migration services, training, software, and documentation
- AXP systems on the Internet

Most of the OpenVMS VAX features and many of the Digital layered products and third-party applications are already available on OpenVMS AXP.

To help protect your investment in Digital products, Digital has introduced new hardware and software programs, such as the Alpha Ready Program, the Easy Alpha Upgrade Program, and the Universal Platform Guarantee Program.

Digital provides a full menu of migration resources so that you can select what is appropriate for your organization. The offerings include an array of migration services which are available from your Digital representative or authorized reseller and from Alpha Migration Centers (also known as Alpha Resource Centers (ARCs)) in many locations worldwide. Migration training, software, documentation, and access to two AXP systems on the Internet are also available.

## 3.1 Availability of Base Operating System Features

Table 3–1 shows selected features in the base operating system that are available with OpenVMS AXP Version 1.5, and the features that will be available in the near future. (In the table, the column heading, Planned, refers to the next several OpenVMS AXP releases.) The list is not complete; it represents the features that are most important to Digital customers. For more information, contact your Digital account representative or authorized reseller.

Table 3–1 Availability of Selected Base Operating System Features on OpenVMS AXP As of June 1993

| Feature | OpenVMS AXP V1.5 | Planned |
|---|---|---|
| Adaptive pool management[1] | Yes | Yes |
| Batch and print queuing system | Yes[2] | Yes |
| Class scheduler system services[1] | No | Yes |
| C2 Security[1] | No | Yes |
| DECdtm | Yes | Yes |
| DECthreads[2] | Yes | Yes |
| Debugger | Yes | Yes |
| EMA base system support | Yes | Yes |
| Extended LAT integration[2] | Yes | Yes |
| Help Message[1] | Yes | Yes |
| Infoserver booting | Yes | Yes |
| ISO 9660 support[1] | Yes | Yes |
| LMF V1.1[2] | Yes | Yes |
| MSCP[3] dynamic load balancing[1] | No | Yes |
| MoveFile function with enhancements[2] | No | Yes |
| Multiple queue manager support[1] | No | Yes |
| Symmetric Multiprocessing (SMP) | Yes | Yes |
| TPU and EVE | Yes | Yes |
| Virtual I/O cache for standalone machines | Yes | Yes |
| Virtual I/O cache clusterwide | No | Yes |
| VMScluster support | Yes | Yes |
| VMScluster support: FDDI interconnect of AXP systems | No | Yes |
| VMScluster support: TMSCP serving by AXP systems | No[4] | Yes |
| X-terminal support | Yes | Yes |
| User-written device driver support | Yes[5] | Yes[6] |

[1] OpenVMS VAX Version 6.0 feature.

[2] OpenVMS VAX Version 5.5 feature.

[3] Mass Storage Control Protocol.

[4] Not available for AXP. However, VAX systems in a VMScluster can serve TMSCP compliant tapes to both VAX and AXP systems in a VMScluster.

[5] Step 1 drivers, available but not recommended.

[6] Step 2 drivers.

## 3.2 Timetable for Digital Layered Products

Many Digital layered products were available for OpenVMS AXP Version 1.0 and many more are available for OpenVMS AXP Version 1.5. Table 3–2 shows some of the layered products that were available on OpenVMS AXP Version 1.0, some that are available with OpenVMS AXP Version 1.5, and some that will be available soon. Not all Digital layered products are listed in Table 3–2. It represents the layered products that are of interest to the largest number of Digital customers.

In the United States and Canada, you can obtain a catalog of available Digital layered products and third-party applications by calling 1-800-DEC-INFO (1-800-332-4636). In other locations, you can obtain the catalog from your Digital account representative or authorized reseller.

Digital layered products are released quarterly. To find out when products not listed in the catalog will be available, contact your Digital account representative or authorized reseller.

Note that the releases are cumulative. For example, the CD–ROM for Digital layered products released in June 1993 contains all the layered products that have been released for OpenVMS AXP. They will all run on OpenVMS AXP Version 1.5.

**Table 3–2 Availability of Selected Digital Layered Products As of June 1993**

| OpenVMS AXP V1.0 | OpenVMS AXP V1.5 | July - Dec. '93 | Jan. - June '94 |
|---|---|---|---|
| **Language and Application Development** | | | |
| DEC C<br>DEC Fortran<br>DECset<br>DXML<br>MACRO-64 | DEC Ada<br>DEC COBOL<br>DEC Pascal<br>OPS5 | C++<br>DATATRIEVE<br>DEC RALLY | DECelx Real Time Tools<br>PL/1<br>DEC BASIC<br>DEC Forté |
| **Middleware** | | | |
| DECwindows Motif (including CDA, Bookreader) | ACAS<br>DEC AVS<br>DECmessageQ<br>DEC GKS<br>DEC PHIGS<br>DQS<br>POLYCENTER Agent<br>Open 3D<br>SQL Services | POSIX[1] | DEC/EDI<br>POSIX[2] |
| **VMSclusters and Associated Software** | | | |
| | VMSclusters<br>DECram for OpenVMS<br>VAXcluster Console System | Volume Shadowing (host-based)[3]<br>RMS Journaling | Disk Striping |

[1] POSIX 1003.1 and 1003.2; XPG3 Base Branding expected during this time.

[2] Final POSIX 1003.2 and 1003.4 standards; XPG4 Component Branding expected during this time.

[3] May be released early as a layered product with some configuration restrictions.

(continued on next page)

**Table 3–2 (Cont.) Availability of Selected Digital Layered Products As of June 1993**

| OpenVMS AXP V1.0 | OpenVMS AXP V1.5 | July - Dec. '93 | Jan. - June '94 |
|---|---|---|---|
| **Transaction Processing and Information Management** | | | |
| FMS | ACMS Desktop Client<br>CDD/Repository<br>DBMS<br>DECforms<br>DEC Rdb<br>SQL Multimedia | ACCESSWORKS<br>ACMS runtime<br>Data Distributor<br>DECobject database | ACMS development<br>Rdb/Expert<br>Instant SQL<br>RTR |
| **End-User Tools** | | | |
| | DEC Notes | | VTX |
| **Other** | | | |
| DECmigrate | PATHWORKS Server [4]<br>POLYCENTER Scheduler<br>POLYCENTER SW DIS[5]<br>DEC X.25 Client | DECamds (driver)<br>POLYCENTER Performance Data Collector | ALL–IN–1 Integrated Office System<br>POLYCENTER File Optimizer<br>PATHWORKS Server (full client support) |

[4] This version supports MS–DOS, OS/2, and windows clients.

[5] POLYCENTER Software Distribution (formerly called Remote System Manager).

## 3.3 Available Third-Party Applications

More than 1,000 third-party applications are currently available for OpenVMS AXP, and the number is growing daily. Some of them are shown in Table 3–3.

In the United States and Canada, you can obtain a catalog listing all the available third-party applications and Digital layered products by calling 1-800-DEC-INFO (1-800-332-4636). In other locations, you can obtain the catalog from your Digital account representative or authorized reseller. To find out when products not listed in the catalog will be available, contact your Digital account representative or authorized reseller.

**Table 3–3 Examples of Available Third-Party Applications As of June 1993**

| Application | Company |
|---|---|
| ACUMATE Version 12 | Kenan Technologies |
| Adabas | Software AG of North America, Inc. |
| ade EKO | Adedata AB. |
| ade INV | Adedata AB. |
| ade T/D | Adedata AB. |

(continued on next page)

Table 3–3 (Cont.) Examples of Available Third-Party Applications As of June 1993

| Application | Company |
|---|---|
| ADINA Version 6.1 | ADINA R&D, INC. |
| ALK-GIAP | AED Graphics GmbH |
| Anvil 5000 Release 3.0 | Manufacturing and Consulting Services, Inc. |
| Application Browser | Hypersoft Europe |
| ARC/INFO Rev. 6.1.1 | Environmental Systems Research Institute, Inc. |
| ASPEN PLUS | Aspen Technology, Inc. |
| Auto. Library System | Dynix, Inc. |
| BEV-PAK | Turn-Key Distribution Systems, Inc. |
| Blood Bank Systems | Antrim Corporation |
| BROKERMAX Version 1.0 | Citymax Integrated Information System Ltd. |
| Business Intelligence Network (BIN) | Henco Software, Inc. |
| Cadim/EDB | EIGNER + PARTNER GmbH |
| CADRA-III | ADRA Systems, Inc. |
| Client Server Interfaces Version 4.9.1 | Sybase, Inc. |
| DADisp | DSP Development Corporation |
| DL Pager | Datalogics, Inc. |
| ECLIPSE | Intera Information Technologies |
| Financial Systems | Antrim Corporation |
| Flexilab System | Sunquest Information Systems, Inc. |
| FLEXIRAD Version 1.0 | Sunquest Information Systems, Inc. |
| FOCUS for Alpha Version 6.5 | Information Builders, Inc. |
| Gembase Open Systems 4GL | Ross Systems, Inc. |
| GENSTAT 5 Version 2.0 | NAG |
| GLIM Version 3.77 | NAG |
| GLIM Version 4.0 | NAG |
| GRAFkit Version 3.3a | Centera Information Systems, Inc. |
| Graphics Language Interpreter Version 4 | Centera Information Systems, Inc. |
| IAS Version 6.4 | CODA, Incorporated |
| IBS-90 | Winter Partners |
| IMPALA | EEC SYSTEMS, INC. |
| Information Support Systems | Antrim Corporation |
| "Integra" Application Management Environment | G.C. McKeown & Co. (UK) Ltd. |
| Integrated Graphics System | Datalogics, Inc. |
| Integration Services | Antrim Corporation |
| IPS Version 4.12 | CODA, Incorporated |
| MANMAN (including Process) | The ASK Companies |

(continued on next page)

Table 3–3 (Cont.) Examples of Available Third-Party Applications As of June 1993

| Application | Company |
|---|---|
| MANTIS Version 2.6 | Cincom Systems, Inc. |
| MAPS | Logica Industry Limited |
| MARGO Version 3 | SEMA GROUP-PROGICIELS |
| Mathematica | Wolfram Research, Inc. |
| Mathematica Version 2.2 | Wolfram Research, Inc. |
| MESSIDOR Version 4 | SEMA GROUP-PROGICIELS |
| Mobile Version 3.2 | KINGSTON_SCL LTD. |
| MultiNet | TGV, Inc. |
| NAG FORTRAN Library Version 15 | NAG |
| NAG Graphics Library Version 4 | NAG |
| NAGWave Fortran 90 Compiler | NAG |
| NATURAL | Software AG of North America, Inc. |
| NETRON/CAP Version 2.07 | NETRON, Inc. |
| NETRON/Client Version 2.07 | NETRON, Inc. |
| Oasys 680x0 Cross Tools | Oasys |
| ORACLE V6 Migration release | Oracle Corporation |
| ORACLE V7 (Developer's release) | Oracle Corporation |
| ORACLE V7 (Production release) | Oracle Corporation |
| Oracle Financials and Human Resources Release 9.4 | Oracle Corporation |
| Oracle Manufacturing Release 9.4 | Oracle Corporation |
| Page Station/X | Datalogics, Inc. |
| PixTex/EFS Version 3.0 | Excalibur Technology |
| PLEIADES HOSPITAL ADMINISTRATION Version 8.7 | SEMA GROUP-PROGICIELS |
| PLEIADES HRM/PRIVATE AND PUBLIC SECTORS Version 8.0 | SEMA GROUP-PROGICIELS |
| PLEIADES LOCAL COMMUNITIES Version 8.7 | SEMA GROUP-PROGICIELS |
| Polyserver | Uniface Int. |
| PowerHouse 4GL | Cognos |
| PROGRESS 4GL/RDBMS Version 6.2 | Progress Software Corporation |
| PROGRESS Application and Development Environment V6.2 | Progress Software Corporation |
| PROGRESS Developer's Toolkit Version 6.2 | Progress Software Corporation |
| PROGRESS FAST TRACK Version 6.2 | Progress Software Corporation |
| PROGRESS RESULTS Query/Reporting Tool Version 6.2 | Progress Software Corporation |

(continued on next page)

Table 3–3 (Cont.) Examples of Available Third-Party Applications As of June 1993

| Application | Company |
|---|---|
| PROMIS Version 5.0 | Promis Systems Corporation |
| Promix Distribution Series | Ross Systems, Inc. |
| Promix Manufacturing Series | Ross Systems, Inc. |
| Renaissance CS Financial Series | Ross Systems, Inc. |
| Renaissance CS Human Resource Series | Ross Systems, Inc. |
| RSC-PR Payroll Version 3.5.3 | Resource Systems Corporation |
| SAP R/3 System Version 1.1 | SAP of America, Inc. |
| SAS System Version 6.0X | SAS Institute, Inc. |
| SmartStar Report Painter | SmartSystems (UK) Ltd. |
| SmartStar Vision | SmartStar Corporation |
| STADEN Package | Medical Research Council |
| STUDENTS+ | J.H. Leskin Associates, Inc. |
| SuperCache | EEC Systems, Inc. |
| SuperDisk | EEC Systems, Inc. |
| SUPRA | Cincom Systems, Inc. |
| Sybase Lifecycle Tools Version 4.9.1 | Sybase, Inc. |
| Sybase SQL Server Version 4.9.1 | Sybase, Inc. |
| Synchrony 3.6 | Henco Software, Inc. |
| TCM-EMS Time Critical Manufacturing System V | Effective Management Systems, Inc. (EMS) |
| TGRAF-X | Grafpoint, Inc. |
| Timeserver | Pilot Software Ltd. |
| TROPOS | Strategic Systems International |
| UniData RDBMS Version 3.0 | Unidata, Inc. |
| Uniface Development Environment | Uniface and Uniface Int. |
| UNIGRAPHICS Version 9.1 | EDS Unigraphics |
| WIN/TCP | Wollongong |
| Wisconsin Sequence Analysis Package Version 7.2.1 | Genetics Computer Group |
| WITNESS (U.S.) Version 5.0 | AT&T ISTEL |

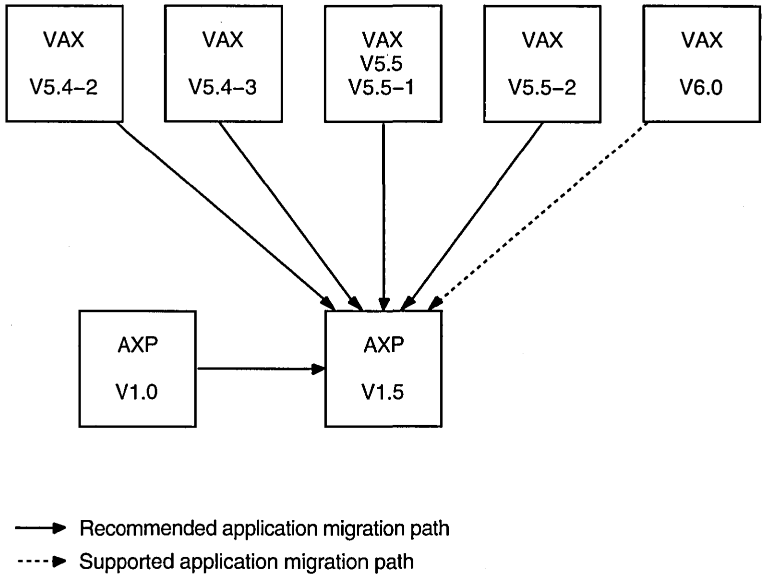
## 3.4 Application Migration Paths

Figure 3–1 shows several paths for application migration from OpenVMS VAX to OpenVMS AXP Version 1.5 systems.

**Note**

In the legend, *Supported* means that most applications will run, but applications that depend on features that are not available in OpenVMS AXP Version 1.5 will not run or will not run correctly.

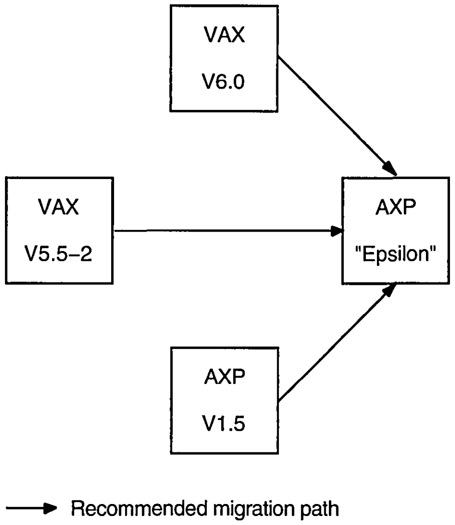
**Figure 3–1 Application Migration to OpenVMS AXP Version 1.5**


VAX
V5.4–2
VAX
V5.4–3
VAX
V5.5
V5.5–1
VAX
V5.5–2
VAX
V6.0
AXP
V1.0
AXP
V1.5
Recommended application migration path
Supported application migration path

ZK–6041A–GE

Figure 3–2 shows several paths that are planned for application migration from OpenVMS VAX to OpenVMS AXP Version "Epsilon."

**Figure 3–2 Application Migration to OpenVMS AXP Version "Epsilon"**



ZK–6040A–GE

## 3.5 Hardware and Software Investment Protection Programs

Digital offers hardware and software programs designed to protect your investment in VAX computers, OpenVMS VAX, OpenVMS AXP, and Digital layered products. The following programs are currently offered:

- **Alpha Ready Program**

  Customers who buy new VAX computers can purchase a simple and cost-effective option to upgrade to an AXP computer within a defined time period. The operating system conversion from OpenVMS VAX to OpenVMS AXP is included in the program.

- **Easy Alpha Upgrade Program**

  Customers who are ready for AXP computers can purchase a simple AXP upgrade from older VAX computers or use the trade-in credit from their existing VAX computers.

- **Universal Platform Guarantee Program**

  Customers can trade in the original AXP operating system licenses for 75% credit (same as standard upgrade policy) towards new licenses on an AXP computer. This gives customers the flexibility of switching to another operating system at a later date.

For Digital layered products, user-based licenses are valid across hardware architectures so no special program has been introduced. New clusterwide and capacity-based licenses are significantly discounted. For more information, contact your Digital account representative or authorized reseller.

## 3.6 Migration Services

Digital customizes the level of service to meet your needs. The migration services available include the following:

- AXP Migration Service Package
- AXP Migration Tools Package
- Orientation Service
- Detailed Analysis
- Migration Support
- Project Planning
- Custom Project Service

The first three services are available from DECdirect (1-800-DIGITAL). The order numbers listed for these services are in effect in the United States. The last four services are tailored to the needs of your organization.

To determine which services are appropriate for you, contact your Digital account representative or authorized reseller or call 1-800-832-6277 (within the United States) or 1-603-884-8990 (from other locations).

### 3.6.1 AXP Migration Service Package

The AXP Migration Service Package includes the following items:

- Compact disc kit, including migration tools and documentation

  The migration tools consist of cross-compilers for several languages (DEC C, DEC Fortran, BLISS, and MACRO–32), a cross-linker, DECmigrate for OpenVMS AXP, and other cross-tools. They enable you to convert, in your familiar programming environment on your OpenVMS VAX system, your existing code to images that can run on an OpenVMS AXP system. If you have the source code for your applications, you can recompile and relink them. If you have only the images, or if a cross-compiler is not available for the language in which a module or entire application is written, you can translate the image.

- Computer-based instruction, *Alpha AXP Architecture Concepts*
- Training credit account
- Telephone support

You can purchase this service package before your AXP hardware arrives and begin your migration effort immediately.

This service is available through DECdirect, U.S. order #YS-ALPAA-OH.

### 3.6.2 AXP Migration Tools Package

The AXP Migration Tools Package is a subset of the AXP Migration Service Package. It contains only the compact disc kit and telephone support. This package is appropriate for customers who have studied the AXP architecture.

This service is available through DECdirect, U.S. order #YS-ALAAA-OH.

### 3.6.3 Orientation Service

The Orientation Service helps you understand the issues involved in migrating an application from OpenVMS VAX to OpenVMS AXP systems. This package is recommended for all Alpha AXP migration customers.

This service is available through DECdirect, U.S. order #QS-ALPAA-CA.

### 3.6.4 Detailed Analysis

The Detailed Analysis service provides a customized methodology for planning the migration of applications from OpenVMS VAX to OpenVMS AXP systems. A Digital consultant conducts a detailed investigation of the customer's application, highlights items that potentially impact the migration effort and schedule, and produces a detailed migration assessment plan. This service is recommended for all customers who would like help with their migration.

### 3.6.5 Migration Support

Migration Support is provided by a Digital migration expert who assists in the migration project. This service includes onsite technical consulting at the level you require.

### 3.6.6 Project Planning

The Project Planning service is provided by a Digital consultant. The consultant provides the following:

- Information that you need to understand the size and scope of a migration project
- Information that you need to develop a detailed migration strategy based on your needs
- Detailed migration project plan

### 3.6.7 Custom Project Service

The Custom Project Service results in a total turnkey migration. It is recommended for customers who want Digital to port their entire application.

### 3.6.8 AMCs/ARCs

Migration services are provided through worldwide Alpha Migration Centers (AMCs), also known as Alpha Resources Centers (ARCs), as shown in Table 3–4, and from Digital's Professional Service Centers. The centers are staffed with migration experts who assist third-party vendors and customers with their porting efforts.

For detailed information on migration services, contact your Digital account representative or authorized reseller or call 800-832-6277 or 603-884-8990.

**Table 3–4 Locations of AMCs/ARCs**

| Europe | North America and Latin America | Asia and Australia |
|---|---|---|
| Austria, Vienna | California, Irvine | Hong Kong |
| Belgium, Brussels | California, Palo Alto | Israel, Herzelia |
| Denmark, Hoersholm | Colorado, Colorado Springs[1] | Japan, Tokyo |
| Finland, Espoo | Georgia, Atlanta | Australia, Rhodes |
| France, Evry | Illinois, Chicago | |
| Germany, Unterfoehring | Michigan, Detroit | |
| Greece, Athens | New Hampshire, Merrimack | |
| Ireland, Galway | New Jersey, Saddle Brook | |
| Italy, Milan | Texas, Houston | |
| Italy, Turin | Washington, D.C. | |
| The Netherlands, Utrecht | Canada, Toronto | |
| Norway, Oslo | Latin America (AMC in Deerfield Beach, FL) | |
| Portugal, Lisbon | | |
| Spain, Madrid | | |
| Sweden, Stockholm | | |
| Switzerland, Duebendorf | | |
| United Kingdom, Reading | | |

[1] There are two AMCs in Colorado Springs; one is a secure AMC, staffed by migration experts with security clearances.

## 3.7 Migration Training

Digital Customer Training offers several seminars and courses to provide migration training to third-party application developers and end users. The first course in the following list of courses is designed for technical or MIS managers, and the others are designed for experienced OpenVMS VAX programmers.

- Alpha AXP Planning Seminar—2 days, EY-L570E-SO
- Migrating HLL Applications to OpenVMS Alpha AXP—3 days, EY-L577E-LO
- Migrating MACRO–32 Applications to OpenVMS AXP—2 days, EY-L578E-LO
- OpenVMS Alpha AXP Step 1 Device Drivers—5 days, EY-L579E-LO

To obtain a schedule and enrollment information in the United States, call 1-800-332-5656. In other locations, contact your Digital account representative or authorized reseller.

## 3.8 Migration Software

In addition to the migration software described in the AXP Migration Service Package, Digital offers DECmigrate for OpenVMS AXP, a Digital layered product. DECmigrate is used for the following purposes:

- To analyze code to determine how easy or difficult it might be to migrate it
- To translate images for which you have no sources or whose native compiler is not yet available on OpenVMS AXP systems

The VAX Environment Software Translator (VEST) component of DECmigrate translates the VAX binary image file into a native AXP image. The translated image runs under the Translated Image Environment (TIE) on an AXP computer. (TIE is a shareable image that is included with the OpenVMS AXP operating system.) Translation does not involve running an OpenVMS VAX image under emulation or interpretation (with certain limited exceptions). Instead, the new OpenVMS AXP image contains Alpha AXP instructions that perform operations identical to those performed by the instructions in the original OpenVMS VAX image.

A translated image generally runs as fast on an AXP computer as the original image runs on a VAX computer. However, since the translated image does not benefit from the optimizing compilers that take full advantage of the Alpha AXP architecture, it will typically run only about 25% to 40% as fast as a native OpenVMS AXP image. The causes for this reduced performance are unaligned data and extensive use of complex VAX instructions.

For more information on image translation and VEST, see *DECmigrate for OpenVMS AXP Systems Translating Images*.

### 3.8.1 Mixing Native AXP and Translated Images

You can mix migration methods among the individual images that comprise an application. An application can also be partially translated as one stage in a migration: this allows the application to run and to be tested on an AXP computer before being completely recompiled. For more information about interoperability of native AXP and translated VAX images within an application, see *Migrating to an OpenVMS AXP System: Recompiling and Relinking Applications*.

## 3.9 Migration Documentation

Digital offers several migration documents. Migration information is also provided in the language user's guides for the DEC compilers. Where differences exist in the DEC compilers between computer systems, those differences are described in the respective language user's guides. In some language guides, such as the *DEC C User's Guide for OpenVMS Systems*, the differences are described in the context of the description of a language element; in other guides, such as the *DEC COBOL User Manual for OpenVMS AXP Systems*, the differences are described in separate appendixes.

The following list describes the migration manuals. Table 3–5 lists their order numbers.

- AXP Migration Kit, which consists of the following documents:
  - *Migrating to an OpenVMS AXP System: Planning for Migration*

    This manual describes the general characteristics of RISC architectures, compares the Alpha AXP architecture to the VAX architecture, and presents an overview of the migration process and a summary of migration tools provided by Digital. The information in this manual is intended to help you define the optimal migration strategy for your application.
  - *Migrating to an OpenVMS AXP System: Recompiling and Relinking Applications*

    This manual provides detailed technical information for programmers who must migrate mid- and high-level language applications to OpenVMS AXP systems. It describes how to set up a development environment to facilitate the migration of applications, helps programmers identify application dependencies on elements of the VAX architecture, and introduces compiler features that help resolve these dependencies. Individual sections of this manual discuss specific application dependencies on VAX architectural features, data porting issues (such as alignment concerns), and the process of migrating VAX shareable images.
  - *Migrating to an OpenVMS AXP System: Porting VAX MACRO Code*

    This manual describes how to use the MACRO–32 compiler for OpenVMS AXP to port VAX MACRO code to an OpenVMS AXP system. It describes the features of the compiler, presents a methodology for porting VAX MACRO code, identifies nonportable coding practices, and recommends alternatives to such practices. The manual also provides detailed descriptions of the compiler qualifiers, directives, built-ins, and the system macros created for porting to an OpenVMS AXP system.
- *DECmigrate for OpenVMS AXP Systems Translating Images*

  This manual describes the VAX Environment Software Translator (VEST) utility, discussed in Section 3.8. It describes how to use VEST to convert most user-mode OpenVMS VAX images to translated images that can run on OpenVMS AXP systems; how to improve the run-time performance of translated images; how to use VEST to trace OpenVMS AXP incompatibilities in an OpenVMS VAX image back to the orginal source files; and how to use VEST to support compatibility among native and translated run-time libraries.

- *OpenVMS AXP Device Support: Creating a Step 1 Driver from an OpenVMS VAX Driver*

  This manual describes how to create a temporary (Step 1) OpenVMS AXP device driver from an existing OpenVMS VAX device driver. (For a description of Step 1 driver support, see Section 1.4.2.3.)

### 3.9.1 How to Order Documentation

The order numbers for the AXP Migration Kit and the migration documents are shown in Table 3–5.

When you purchase an OpenVMS AXP media kit, you receive the AXP Migration Kit in Bookreader format (DECW$BOOK) on the compact disc. When you purchase the printed OpenVMS AXP Standard Documentation Set, the AXP Migration Kit is included. You can also order this kit or any of its manuals separately. *DECmigrate for OpenVMS AXP Systems Translating Images* in Bookreader format accompanies the optional layered product, DECmigrate for OpenVMS AXP. To obtain a printed copy, you must order it separately.

**Table 3–5 Documentation Order Numbers**

| Manual or Kit | Order Number |
|---|---|
| AXP Migration Kit | QA-MT1AG-GZ.1.5 |
| *Migrating to an OpenVMS AXP System: Planning for Migration* | AA-PV62A-TE |
| *Migrating to an OpenVMS AXP System: Recompiling and Relinking Applications* | AA-PV63A-TE |
| *Migrating to an OpenVMS AXP System: Porting VAX MACRO Code* | AA-PV64A-TE |
| *DECmigrate for OpenVMS AXP Systems Translating Images* | AA-PSGMB-TE |
| *OpenVMS AXP Device Support: Creating a Step 1 Driver from an OpenVMS VAX Driver* | AA-PV73A-TE |

The documentation can be ordered by telephone, by mail, and, in the United States, from the Electronic Store. The directions follow.

#### Telephone Orders

| | |
|---|---|
| **Continental USA, Alaska, or Hawaii** | Call 800-DIGITAL<br>800-344-4825<br>FAX: (603) 884-5597 |
| **Puerto Rico** | Call 809-781-0505<br>FAX: (809) 749-8377 |
| **Canada** | Call 800-267-6215<br>FAX: (613) 592-1946 |
| **International** | Call local Digital subsidiary or approved distributor |

### Direct Mail Orders

**Continental USA,
Alaska, or Hawaii**

Digital Equipment Corporation
P.O. Box CS2008
Nashua, New Hampshire 03061

**Puerto Rico**

Digital Equipment Caribbean, Inc.
3 Digital Plaza, 1st Street
Suite 200
Metro Office Park
San Juan, Puerto Rico 00920

**Canada**

Digital Equipment of Canada Ltd.
Attn: DECdirect Sales
100 Herzberg Road
Kanata, Ontario, Canada K2K 2A6

**International**

Contact local Digital subsidiary or approved distributor

### Electronic Orders, USA Only

To place an order from the Electronic Store, dial 800-234-1998 using a 2400- or 9600-baud modem. If you need help using the Electronic Store, call 800-DIGITAL (800-344-4825).

## 3.10 AXP Systems on the Internet

As a service to the Internet community, Digital has made available two DEC 4000 AXP systems. These systems are available for evaluating the Alpha AXP architecture and for testing the features of the supporting OpenVMS AXP and DEC OSF/1 for AXP operating systems, compilers, tools, and utilities.

Application developers who have access to the Internet can use these two systems to test, qualify, or port their software for the Alpha AXP architecture. Other Internet users interested in Alpha AXP computing are invited to log in and evaluate these systems.

One DEC 4000 AXP system (Internet address: axpvms.pa.dec.com) has the OpenVMS AXP V1.5 operating system installed. The other DEC 4000 AXP system (Internet address: axposf.pa.dec.com) is running the DEC OSF/1 for AXP V1.2 UNIX operating system. These systems can be reached either via telnet or rlogin.

To register for an account, Internet users connect to the desired machine, log in as axpguest (no password), and answer the short qualifying questionnaire. Users are asked to read all information in the motd/login banner and comply with all rules for machine usage/restrictions.

Users with questions about their accounts should send mail to Internet address: axpvms-system@pa.dec.com for the OpenVMS AXP system and to Internet address: axposf-root@pa.dec.com for the DEC OSF/1 for AXP system.

# 4 Ensuring the Portability of Applications

This chapter describes:

- How to assess the portability of an application
- OpenVMS AXP features that contribute to portability
- Differences in the OpenVMS AXP programming environment
- Guidelines for new program development for OpenVMS VAX and OpenVMS AXP

In general, if your application is written in a high-level programming language, you should be able to run it on an AXP system with a minimum amount of effort. High-level languages insulate applications from dependence on the underlying machine architecture. In addition, for the most part, the programming environment on AXP systems duplicates the programming environment on VAX systems. Using native AXP versions of the language compilers and the Linker utility (linker), you can recompile and relink the source files that make up your application to produce a native AXP image.

If your application is written in VAX MACRO, you may be able to run it on an AXP system with a minimum amount of effort, although it is more likely to contain some dependencies on the underlying VAX architecture, some of which may require your intervention.

## 4.1 How to Assess the Portability of an Application

The portability of an application depends on the language in which it is written, the amount of nonstandard coding practices used by the programmer, the number of architectural dependencies that it contains, and whether a compiler is available for the language in which the application is written. While it is possible to introduce architectural dependencies in applications written in high-level languages, they are more likely to occur in applications written in mid- and low-level languages.

Privileged applications, which run in inner modes or at elevated interrupt priority levels (IPLs), may require significant changes because of assumptions incorporated in the code about the internal operation of the operating system. Typically, such applications also require significant changes between major releases of the OpenVMS VAX operating system.

Recently, Digital introduced new DEC compilers that use a common back-end technology. They produce highly optimized code. It is likely that the applications that you might want to move to an OpenVMS AXP system were compiled using the earlier VAX compilers.

To assess the portability of an application, consider the following:

- The application's dependencies on the VAX architecture
- The differences between the VAX and DEC language compilers
- The diagnostic features of the compilers and DECmigrate for OpenVMS AXP

You may also need to identify nonstandard coding practices. They are generally more common in code written in lower-level languages, such as VAX MACRO. For information about such practices for VAX MACRO, refer to *Migrating to an OpenVMS AXP System: Porting VAX MACRO Code.*

### 4.1.1 Identifying Dependencies on the VAX Architecture in Your Application

Even if your application recompiles successfully with a compiler that generates native AXP code, it may still contain subtle dependencies on the VAX architecture. The OpenVMS AXP operating system has been designed to provide a high degree of compatibility with OpenVMS VAX; however, the fundamental differences between the VAX and AXP architectures can create certain problems for applications that depend on certain VAX architectural features. The following list highlights those areas of your application you should examine.

- Check the data declarations contained in your application. The high-level language data types you selected to represent data items on a VAX system may not be the best choice on an AXP system. In particular, consider the following:
  - **Data packing**—Applications on VAX systems typically use the smallest available data type to represent a data item to achieve efficient use of memory resources. For various reasons, using larger data types may be more efficient on AXP systems. For example, unaligned data can take 100 times longer to process than aligned.
  - **Data-type selection**—The Alpha AXP architecture supports most of the VAX native data types; however, certain VAX data types, such as the H_float floating-point data type, are not supported. Check to see if your application depends on the size or bit representation of an underlying native data type.
  - **Shared access to data**—Check any writeable data item that is accessed by multiple threads of execution. The VAX architecture includes instructions that can perform certain complex operations, such as incrementing a variable, that appear as a single, noninterruptable operation to other threads of execution. The Alpha AXP architecture is a load-store architecture that does not support atomic memory-to-memory modifications so different program constructs may be required.

    In addition, the VAX architecture supports instructions that can manipulate byte- and word-sized data in a single noninterruptable operation. The Alpha AXP architecture supports noninterruptable access only to aligned longword- or aligned quadword-sized data.
  - **Buffer size**—Your application may determine the size of certain data buffers based on the VAX page size. Different implementations of the Alpha AXP architecture can support 8K, 16K, 32K, or 64K byte pages. Search your application for the text strings "512" and "511" (or the hexadecimal equivalents, "200" and "1FF") to find dependencies on the VAX page size.

- Check any condition handlers your application may include. While the condition handling facility on AXP systems is functionally equivalent to the VAX condition handling facility, certain aspects of the facility have changed, such as the format of the mechanism array. In addition, the way in which arithmetic exceptions are reported has changed.
- Check for dependence on the AST parameter list. While the AST parameter list on AXP systems has the same format as on VAX systems, only the AST parameter field can be used. The other fields in the AST parameter list (contents of R0, R1, program counter [PC], and processor status [PS]) are provided for compatibility only and have no subsequent use after the AST procedure exits. For example, on OpenVMS VAX systems, some user-written AST procedures are designed to change one or more of the values in the other fields in the AST parameter list so that these new values take effect upon completion of the AST procedure. Because ASTs are handled differently on OpenVMS AXP, such changes by the AST procedure to the other fields in the AST parameter list have no effect. Anything an AST procedure writes to the last four parameters on an AXP computer is lost when the AST procedure exits.

For more information, see *Migrating to an OpenVMS AXP System: Recompiling and Relinking Applications* for applications written in mid- and high-level languages and the user's guides for the particular language. For applications written in VAX MACRO, see *Migrating to an OpenVMS AXP System: Porting VAX MACRO Code.*

### 4.1.2 Compiler Differences

Compiler differences can exist for two reasons: the changes from the VAX version to the DEC version and the changes from one computer architecture to another. The compilers available on OpenVMS AXP systems are intended to be compatible with their counterparts on OpenVMS VAX systems. The languages conform to language standards and include support for most OpenVMS VAX language extensions. The compilers produce output files with the same default file types as they do on OpenVMS VAX systems, such as .OBJ for an object module.

Note, however, that some features supported by the compilers on OpenVMS VAX systems may not be available in the same compiler on OpenVMS AXP systems. For example, the OpenVMS AXP run-time libraries do not contain the routine LIB$ESTABLISH, which the OpenVMS VAX run-time libraries contain. Due to the nature of the Alpha AXP Calling Standard, setting up condition handlers is done by compilers.

For those programs that need to dynamically establish condition handlers, some AXP languages give special treatment for apparent calls to LIB$ESTABLISH and generate the appropriate code without actually calling an RTL routine. The following languages support LIB$ESTABLISH semantics in a compatible fashion with the corresponding VAX language:

- DEC C and DEC C++

  Although DEC C and DEC C++ on OpenVMS AXP treat LIB$ESTABLISH as a built-in function, the use of LIB$ESTABLISH is not recommended on OpenVMS VAX or OpenVMS AXP systems. C and C++ programmers should call VAXC$ESTABLISH instead of LIB$ESTABLISH (VAXC$ESTABLISH is a built-in function on DEC C and DEC C++ OpenVMS AXP).

- DEC Fortran

  DEC Fortran allows declarations to LIB$ESTABLISH and converts them to DEC Fortran RTL specific entry points.
- DEC Pascal

  DEC Pascal provides the built-in routines, ESTABLISH and REVERT, to use in place of LIB$ESTABLISH. If you declare and try to use LIB$ESTABLISH, you will get a compile-time warning.
- MACRO-32

  The MACRO-32 compiler will attempt to call LIB$ESTABLISH if it is contained in the source code.

  If MACRO-32 programs establish dynamic handlers by storing a routine address at 0(FP), they will work correctly when compiled on an OpenVMS AXP system. However, you cannot set the condition handler address from within a JSB (Jump to Subroutine) routine, only from within a CALL_ENTRY routine.

In addition, some compilers on OpenVMS AXP systems support new features not supported by their counterparts on OpenVMS VAX systems. To provide compatibility, some compilers support compatibility modes. For example, the DEC C for OpenVMS AXP compiler supports a VAX C compatibility mode that is invoked by specifying the /STANDARD=VAXC qualifier.

In some cases, the compatibility is limited. For example, VAX C implements built-in functions that allow access to special VAX hardware features. Since the hardware architecture of VAX computers differs from AXP computers, these built-ins are not available on DEC C for OpenVMS AXP even when the /STANDARD=VAXC qualifier is used.

If you are recompiling VAX C code, either an entire application or one or more modules, you will want to pay particular attention to any external symbols that it contains. Unlike the VAX C compiler which supports one external symbol model, the DEC C compiler supports four models. The default external symbol that is produced by the DEC C compiler is not the same as the single VAX C external symbol.

Furthermore, when you link such code, due to changes in the linker, if you did not specify the /SHARE qualifier when you recompiled the C code module, you will need to specify a related linker qualifier.

For more information about the differences between DEC C and VAX C and the differences between DEC Fortran and VAX FORTRAN, refer to *Migrating to an OpenVMS AXP System: Recompiling and Relinking Applications*. For more information about the compiler differences for each language, refer to its documentation, especially, the user's guides and the release notes. For more information about the linker, refer to the *OpenVMS Linker Utility Manual*.

## 4.2 OpenVMS AXP Features That Contribute to Portability

The OpenVMS AXP operating system and compilers include features that contribute to portability. DECmigrate for OpenVMS AXP Systems is a migration tool with diagnostic features. Some of the OpenVMS AXP operating system and compiler features are primarily diagnostic while others compensate for architectural dependencies.

### 4.2.1 Diagnostic Features

The compilers can produce messages that are very useful for identifying potential porting problems. For example, the MACRO–32 compiler provides the /FLAG qualifier with 10 options. Depending on which options you include, the compiler reports all unaligned stack and memory references, any run-time code generation (such as self-modifying code), branches between routines, or several other conditions.

The DEC Fortran compiler qualifer, /CHECK, produces messages about any of the various options you specify.

A component of DECmigrate, the VAX Environment Software Translator (VEST) can be used to analyze images. It can identify specific incompatibilities for an AXP computer. Depending on the type of incompatibility, you can choose to specify a compiler qualifier that will compensate for the problem or make changes to the code. For more information, see *DECmigrate for OpenVMS AXP Systems Translating Images*

### 4.2.2 Features That Minimize Architectural Differences

The compilers can compensate for some architectural dependencies that may exist in your code. For example, the MACRO–32 compiler provides the /PRESERVE qualifier that can preserve granularity or atomicity or both.

The DEC C compiler provides a header file that defines macros for each data type. These macros map a generic data-type name, such as int64, to the machine-specific data type, such as *-64*. For example, if you must have a data type that is 64 bits long, use the int64 macro.

Review the documentation for your compiler to become familiar with all its features that facilitate migration.

### 4.2.3 PALcode

The Alpha AXP architecture does not favor a particular operating system. To accommodate different operating systems, it enables the creation of privileged architecture library code (PALcode).

Furthermore, certain OpenVMS AXP compilers, such as C and the MACRO-32 compiler, provide PALcode built-ins that supplement the instructions available in the Alpha AXP instruction set. For example, the MACRO-32 compiler provides built-ins that emulate those VAX instructions for which there are no Alpha AXP equivalents.

PALcode can be used to access internal hardware registers and physical memory. PALcode can provide direct correspondence of physical and virtual memory. For more information about PALcode, see the *Alpha Architecture Reference Manual.*

## 4.3 Differences in the OpenVMS AXP Programming Environment

Additional differences in the OpenVMS AXP program development environment are discussed in this section.

### 4.3.1 Linker

Once you successfully recompile your source files, you must relink your application to create a native AXP image. The linker produces output files with the same file types as on current VAX systems. For example, by default, the linker uses the file type .EXE to identify image files.

Because the way in which you perform certain linking tasks is different on OpenVMS AXP systems, you will probably need to modify the LINK command used to build your application. The following list describes some of these linker changes that may affect your application's build procedure. See the *OpenVMS Linker Utility Manual* for more information.

- **Declaring universal symbols in shareable images**—If your application creates shareable images, your application build procedure probably includes a transfer vector file, written in VAX MACRO, in which you declare the universal symbols in the shareable image. On OpenVMS AXP systems, instead of creating a transfer vector file, you must declare universal symbols in a linker options file by specifying the SYMBOL_VECTOR= option.
- **Linking against the OpenVMS executive**—On OpenVMS VAX systems, you link against the OpenVMS executive by including the system symbol table file (SYS.STB) in your build procedure. On OpenVMS AXP systems, you link against the OpenVMS executive by specifying the /SYSEXE qualifier.
- **Optimizing the performance of images**—On OpenVMS AXP systems, the linker can perform certain optimizations that can improve the performance of the images it creates. In addition, the linker can create shareable images that can be installed as resident images, another performance enhancement.
- **Processing shareable images implicitly**—On OpenVMS VAX systems, when you specify a shareable image in a link operation, the linker also processes all the shareable images to which that shareable image was linked. On OpenVMS AXP systems, to include these shareable images in your build procedure, you must explicitly specify them.

The linker supports several qualifiers and options, listed in Table 4–1, that are specific to OpenVMS AXP systems. Linker qualifiers supported on OpenVMS VAX systems that are not supported by the linker on OpenVMS AXP systems are also included in Table 4–1.

**Table 4–1 Linker Qualifiers and Options Specific to OpenVMS AXP Systems**

| Qualifiers | Description |
|---|---|
| /DEMAND_ZERO | Controls how the linker creates demand-zero image sections. |
| /GST | Directs the linker to create a global symbol table (GST) for a shareable image (the default). More typically specified as /NOGST when used to create shareable images for run-time kits. |
| /INFORMATIONALS | Directs the linker to output informational messages during a link operation (the default). More typically specified as /NOINFORMATIONALS to suppress these messages. |

(continued on next page)

Table 4–1 (Cont.) Linker Qualifiers and Options Specific to OpenVMS AXP Systems

| Qualifiers | Description |
|---|---|
| /NATIVE_ONLY | Directs the linker to *not* pass along the procedure signature block (PSB) information, created by the compilers, in the image it is creating (the default).<br>If /NONATIVE_ONLY is specified during linking, the image activator uses the PSB information, if any, provided in the object modules specified as input files to the link operation to invoke jacket routines. Jacket routines are necessary to allow native AXP images to work with translated VAX images. |
| /REPLACE | Directs the linker to perform certain optimizations that can improve the performance of the image it is creating, when requested to do so by the compilers (the default). |
| /SECTION_BINDING | Directs the linker to create a shareable image that can be installed as a resident image. |
| /SYSEXE | Directs the linker to process the OpenVMS executive image (SYS$BASE_IMAGE.EXE) to resolve symbols left unresolved in a link operation. |

| Options | Description |
|---|---|
| BASE= option | Not supported on OpenVMS AXP systems. |
| DZRO_MIN= option | Not supported on OpenVMS AXP systems. |
| ISD_MAX= option | Not supported on OpenVMS AXP systems. |
| SYMBOL_TABLE= option | Directs the linker to include global symbols as well as universal symbols in the symbol table file associated with a shareable image. By default, the linker includes only universal symbols. |
| SYMBOL_VECTOR= option | Used to declare universal symbols in AXP shareable images. |
| UNIVERSAL= option | Not supported on OpenVMS AXP systems. |

### 4.3.2 MACRO–32 Compiler for OpenVMS AXP

The MACRO–32 Compiler for OpenVMS AXP is used to convert existing VAX MACRO code into machine code that runs on OpenVMS AXP systems. It is included with OpenVMS AXP for that purpose.

While some VAX MACRO code can be compiled without any changes, most code modules will require the addition of entry point directives. Many code modules will require other changes as well.

The compiler generates code that is optimized for OpenVMS AXP systems, but many features of VAX MACRO that provide the programmer with a high level of control make it more difficult to generate optimal code for OpenVMS AXP systems. For new program development for OpenVMS AXP, Digital recommends the use of mid- and high-level languages. For more information on the MACRO–32 compiler, see *Migrating to an OpenVMS AXP System: Porting VAX MACRO Code.*

### 4.3.3 MACRO–64 Assembler for OpenVMS AXP

The MACRO–64 Assembler for OpenVMS AXP systems is the native assembler for all AXP computers. Unlike the VAX MACRO assembler which is included with the OpenVMS VAX operating system, the MACRO-64 assembler is not included with the OpenVMS AXP operating system. It can be purchased separately. In general, the mid- and high-level language compilers generate higher performance code for OpenVMS AXP systems than the MACRO-64 assembler. Therefore, Digital recommends you use mid- and high-level compilers for new program development for OpenVMS AXP systems. For more information about the MACRO-64 assembler, see *MACRO-64 Assembler for OpenVMS AXP Systems Reference Manual.*

### 4.3.4 OpenVMS Debugger

On OpenVMS AXP systems you can use the debugger with programs written in the following DEC languages:

- Ada
- C
- C++
- COBOL
- Fortran
- MACRO-32 (compiled with the MACRO-32 compiler)
- MACRO-64
- Pascal

The OpenVMS Debugger includes several features that address the architectural differences of OpenVMS AXP code. These enable you to more easily debug code that you are porting to OpenVMS AXP systems. For example, you can use the /UNALIGNED_DATA qualifier with the SET command to cause the debugger to break directly after any instruction that accesses unaligned data, such as after a load word instruction that accesses data that is not on a word boundary. You can use the /RETURN qualifier with the SET command for any routine. It is not limited to routines called with a CALLS or CALLG instruction as it is on an OpenVMS VAX system. For more information about features specific to OpenVMS AXP systems, see the *OpenVMS Debugger Manual.*

### 4.3.5 Delta/XDelta Debugger

The Delta/XDelta Debugger (DELTA/XDELTA), running on OpenVMS AXP systems, provides enhancements to existing commands and several new commands necessitated by the Alpha AXP architecture. The enhancements include the display of base registers in decimal instead of hexadecimal notation and the ability to look at the internal process identification (PID) number of another process. The new commands include ;Q, used to validate queues, and ;I, used to locate and display information about the current main image. For DELTA, the ;I command can also display information about all shareable images activated by the current main image. For more information about how the Delta/XDelta Debugger operates on OpenVMS AXP systems, see the *OpenVMS Delta/XDelta Debugger Manual.*

### 4.3.6 System Dump Analyzer

The System Dump Analyzer (SDA) utility on OpenVMS AXP systems is almost identical to the utility provided on OpenVMS VAX systems. Most commands, qualifiers, and displays are identical. There are a few additional commands and qualifiers. Some displays have been adapted to show information specific to OpenVMS AXP systems, such as processor registers and data structures.

While the SDA interface has changed only slightly, the contents of VAX and AXP dump files and the entire process of analyzing a system crash from a dump differ significantly between the two computers. The AXP execution paths leave more complex structures and patterns on the stack than VAX execution paths do.

To use SDA on a VAX computer, you must first familiarize yourself with the OpenVMS calling standard for VAX systems. Similarly, to use SDA on an AXP system, you must familiarize yourself with the OpenVMS calling standard for AXP systems before you can decipher the pattern of a crash on the stack.

The changes to SDA include the following:

- The displays of the SHOW CRASH and SHOW STACK commands contain additional information that make debugging fatal system exception bugchecks simpler.
- The SHOW EXEC command display includes additional information about executive images if they were loaded **sliced**. Slicing is a function performed by the executive image loader for executive images and by the OpenVMS Install utility for user-mode images. Slicing an executive image (or a user-mode image) greatly improves performance by reducing contention for the limited number of translation buffer entries.
- The MAP command, a new SDA command, is provided so you can map a sliced executive image to a map file.
- A new symbol, FPCR, has been added to the symbol table. This symbol represents a floating point register.

### 4.3.7 Mathematics Libraries

Mathematical applications using the standard VMS call interface to the OpenVMS Mathematics (MTH$) Run-Time Library (RTL) need not change their calls to MTH$ routines when migrating to an OpenVMS AXP system. Jacket routines are provided that map MTH$ routines to their math$ counterparts in the Digital Portable Mathematics Library (DPML) for OpenVMS AXP systems. However, there is no support in the DPML for calls made to JSB routine entry points and vector routines. Note that DPML routines are different from those in the OpenVMS MTH$ RTL and you should expect to see small differences in the precision of the mathematical results.

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 or DEC C) rather than using the call interface. Significantly higher performance and accuracy are also available to you with DPML routines.

For more information about the DPML routines, see the *DPML, Digital Portable Mathematics Library* manual.

### 4.3.8 Determining the Host Architecture

Your application may need to determine whether it is running on an OpenVMS VAX system or an AXP system. From within your program, you can obtain this information by calling the $GETSYI system service (or the LIB$GETSYI RTL routine), specifying the ARCH_TYPE item code. When your application is running on a VAX computer, the $GETSYI system service returns the value 1. When your application is running on an AXP computer, the $GETSYI system service returns the value 2.

Example 4–1 illustrates how to determine the host architecture in a DCL command procedure by calling the F$GETSYI lexical function and specifying the ARCH_TYPE item code. For an example of calling the $GETSYI system service in an application to determine the page size of an AXP computer, see *Migrating to an OpenVMS AXP System: Recompiling and Relinking Applications.*

**Example 4–1 Using the ARCH_TYPE Keyword to Determine Architecture Type**

```
$! Determine architecture type
$ type_symbol = f$getsyi("arch_type")
$ if type_symbol .eq. 1 then goto ON_VAX
$ if type_symbol .eq. 2 then goto ON_AXP
$ ON_VAX:
$ !
$ ! Do VAX-specific processing
$ !
$ exit
$ ON_AXP:
$ !
$ ! Do AXP-specific processing
$ !
$ exit
```

Note, however, that the ARCH_TYPE item code is available only on VAX computers running OpenVMS VAX Version 5.5 or later. If your application needs to determine the host architecture for earlier versions of the operating system, use one of the other $GETSYI system service item codes listed in Table 4–2.

**Table 4–2 $GETSYI Item Codes That Specify Host Architecture**

| Keyword | Usage |
|---|---|
| ARCH_TYPE | Returns 1 on a VAX computer; returns 2 on an AXP computer. Supported on AXP computers and on VAX computers running OpenVMS VAX Version 5.5 or a later version. |
| ARCH_NAME | Returns text string "VAX" on VAX computers and text string "Alpha" on AXP computers. Supported on AXP computers and on VAX computers running OpenVMS VAX Version 5.5 or a later version. |
| HW_MODEL | Returns an integer that identifies a particular hardware model. All values equal to or larger than 1024 identify AXP computers. |
| CPU | Returns an integer that identifies a particular CPU. The value 128 identifies a system as "not a VAX." This code is supported on much earlier versions of VMS than the ARCH_TYPE and ARCH_NAME codes. |

### 4.3.9 Uncovering Latent Bugs

Despite your best efforts, and following all the previous suggestions, you may encounter bugs that were in your program all along, but never caused a problem on an OpenVMS VAX system. For example, a failure to initialize some variable in your program might have been benign on a VAX computer but could produce an arithmetic exception on an AXP computer. The same could be true moving between any other two architectures, since the available instructions and the way compilers optimize them is bound to change. There is no magic answer for bugs that have been in hiding, but you should test your programs after porting them before making them available to other users.

## 4.4 Application Compatibility with Future OpenVMS AXP Releases

Nonprivileged programs that run on OpenVMS AXP systems will continue to run unmodified on future OpenVMS AXP releases. As on OpenVMS VAX systems, you will need to recompile and relink privileged programs to run on future OpenVMS AXP releases.

## 4.5 Guidelines for New Program Development

The following guidelines are provided to facilitate the development of programs intended to run on both OpenVMS VAX systems and OpenVMS AXP systems:

1. Familiarize yourself with the changes and the new features of the DEC compiler that you will use. For example:
   - If you are writing any C code with external symbols, familiarize yourself with the changes in DEC C to coding external symbols and compiling such code.
   - If your program needs to dynamically establish condition handlers, you may need to make some changes to your code, as described in Section 4.1.2.
2. Write your code in mid- or high-level languages whenever possible.
3. Follow good programming practices for program modularity.
4. Avoid creating any VAX architectural dependencies in your code. One area that can be troublesome is that of shared memory. The VAX architecture makes certain implicit guarantees about synchronization. If a program requires the same synchronization on the Alpha AXP architecture, it must request it explicitly. This potential problem and others are briefly described in Section 4.1.1. For more information, see *Migrating to an OpenVMS AXP System: Recompiling and Relinking Applications* and the user's guide for the compiler you will use.

   If you cannot avoid relying on the VAX architecture for one or more operations, conditionalize your code for each architecture as described in Section 4.3.8.
5. Familiarize yourself with the differences between the OpenVMS VAX Linker and the OpenVMS AXP Linker. Some of them are described in Section 4.3.1.
6. Examine your OpenVMS VAX build procedures and note what changes may be necessary for recompiling and relinking on an OpenVMS AXP system.
7. Do not use OpenVMS VAX features that are not yet supported by the OpenVMS AXP operating system. If in doubt, check with your Digital account representative or authorized reseller.

8. Use at least aligned longwords for in-memory data structures wherever possible. This has always been more efficient on VAX computers. On AXP computers, the performance difference becomes even greater.

9. Consider buying the AXP Migration Tools Package. (Telephone support is also included.) With this package, you can start to migrate your code before you buy an OpenVMS AXP system. By using the cross-compilers, cross-linker, and other programming tools on an OpenVMS VAX system, you can convert existing code into images that will execute on OpenVMS AXP systems. The AXP Migration Tools package is described in Section 3.6.

# Index

# NOTES

# NOTES

# NOTES

# NOTES

# NOTES

## NOTES

# NOTES

# NOTES

# NOTES

# NOTES

# 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<br>Phone: 800-DIGITAL<br>(800-344-4825)<br>FAX: (603) 884-5597 | Digital Equipment Corporation<br>P.O. Box CS2008<br>Nashua, NH 03061 |
| Puerto Rico | Phone: (809) 781-0505<br>FAX: (809) 749-8377 | Digital Equipment Caribbean, Inc.<br>3 Digital Plaza, 1st Street<br>Suite 200<br>Metro Office Park<br>San Juan, Puerto Rico 00920 |
| Canada | Phone: 800-267-6215<br>FAX: (613) 592-1946 | Digital Equipment of Canada Ltd.<br>100 Herzberg Road<br>Kanata, Ontario, Canada K2K 2A6<br>Attn: DECdirect Sales |
| International | ———— | Local Digital subsidiary or<br>approved distributor |
| Internal Orders[1]<br>(for software<br>documentation) | DTN: 241-3023<br>(508) 874-3023 | Software Supply Business (SSB)<br>Digital Equipment Corporation<br>1 Digital Drive<br>Westminster, MA 01473 |
| Internal Orders<br>(for hardware<br>documentation) | DTN: 234-4325<br>(508) 351-4325<br>FAX: (508) 351-4467 | Publishing & Circulation Services<br>Digital Equipment Corporation<br>NR02-2<br>444 Whitney Street<br>Northboro, MA 01532 |

[1] Call to request an Internal Software Order Form (EN–01740–07).

# Reader's Comments

**OpenVMS Compatibility Between VAX and AXP**
AA–PYQ4A–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 ______________

--- Do Not Tear - Fold Here and Tape ---

digital™

No Postage
Necessary
if Mailed
in the
United States

**BUSINESS REPLY MAIL**

FIRST CLASS PERMIT NO. 33 MAYNARD MASS.

POSTAGE WILL BE PAID BY ADDRESSEE

DIGITAL EQUIPMENT CORPORATION
OpenVMS Documentation
110 SPIT BROOK ROAD ZKO3–4/U08

NASHUA, NH 03062–2642

--- Do Not Tear - Fold Here ---