
A Comparison of System Management on OpenVMS AXP and OpenVMS VAX

Order Number: AA-PV71B-TE

March 1994

This manual compares system management on OpenVMS AXP and OpenVMS VAX operating systems. It is intended for experienced system managers who need to learn quickly how specific tasks differ or remain the same on the two systems. The comparison is between OpenVMS AXP Version 6.1 and the following releases: VMS Version 5.4, VMS Version 5.5, OpenVMS VAX Version 6.0, and OpenVMS VAX Version 6.1.

Revision/Update Information: This manual supersedes
*A Comparison of System Management
on OpenVMS AXP and OpenVMS VAX*,
OpenVMS AXP Version 1.5.

Software Version: OpenVMS AXP Version 6.1

**Digital Equipment Corporation
Maynard, Massachusetts**

March 1994

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

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

© Digital Equipment Corporation 1994. All rights reserved.

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

The following are trademarks of Digital Equipment Corporation: Alpha AXP, AXP, Bookreader, CI, DDCMP, DEC, DECamds, DECdtm, DECnet, DECnet/OSI, DECram, DECterm, DECtrace, DECwindows, DEQNA, Digital, DSA, HSC, HSC70, InfoServer, LASTport, LAT, MicroVAX, MicroVAX II, MSCP, OpenVMS, PATHWORKS, Q-bus, RA, TMSCP, TURBOchannel, ULTRIX, UNIBUS, VAX, VAX DOCUMENT, VAX MACRO, VAX 6000, VAX 8300, VAX 9000, VAXcluster, VAXserver, VAXstation, VMS, VMS RMS, VMScluster, XMI, XUI, the AXP logo, and the DIGITAL logo.

The following are third-party trademarks:

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

Microsoft, MS, and MS-DOS are registered trademarks of Microsoft Corporation.

Open Software Foundation is a trademark of the Open Software Foundation, Inc.

PostScript is a registered trademark of Adobe Systems Incorporated.

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

ZK6010

This document is available on CD-ROM.

This document was prepared using VAX DOCUMENT Version 2.1.

Send Us Your Comments

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

You can send comments to us in the following ways:

- Internet electronic mail: `OPENVMSDOC@ZKO.MTS.DEC.COM`
- Fax: 603-881-0120 Attn: OpenVMS Documentation, ZK03-4/U08
- A completed Reader's Comments form (postage paid, if mailed in the United States), or a letter, via the postal service. Two Reader's Comments forms are located at the back of each printed OpenVMS manual. Please send letters and forms to:

Digital Equipment Corporation
Information Design and Consulting
OpenVMS Documentation
110 Spit Brook Road, ZK03-4/U08
Nashua, NH 03062-2698
USA

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

Thank you.

Contents

Preface	ix
1 Overview of OpenVMS AXP System Management	
1.1 Reduced Number of System Management Differences	1-2
1.2 Why Do Some Differences Still Exist?	1-2
1.2.1 Different Page Size	1-2
1.2.2 A New Way to Perform Standalone Backups and Other Tasks	1-4
1.2.3 DECEvent Event Management Utility	1-4
1.2.4 I/O Subsystem Configuration Commands in SYSMAN	1-4
1.2.5 MONITOR POOL Command Not Necessary	1-5
1.2.6 Changes in OpenVMS File Names	1-5
1.2.7 Unsupported Features in DECnet for OpenVMS AXP	1-5
1.2.8 Unsupported Optional Layered Products	1-5
2 System Setup Tasks	
2.1 Setup Tasks That Are the Same	2-1
2.2 Setup Tasks That Are Different	2-5
2.2.1 New Way to Back Up System Disks	2-6
2.2.2 VMSINSTAL and the POLYCENTER Software Installation Utility	2-6
2.2.3 Planning for and Managing Common Object and Image File Extensions	2-7
2.2.4 BOOT Console Command	2-10
2.2.4.1 DBG_INIT and USER_MSGS Boot Flags	2-11
2.2.5 CONSCOPY.COM Command Procedure Not Available	2-12
2.2.6 Use of the License Management Facility (LMF)	2-13
2.2.7 PAK Name Difference Using DECnet for OpenVMS AXP	2-14
2.2.8 PAK Name Difference for VMSclusters and VAXclusters	2-14
2.2.9 SYSGEN Utility and System Parameters	2-15
2.2.9.1 MULTIPROCESSING System Parameter	2-15
2.2.9.2 PHYSICAL_MEMORY System Parameter	2-15
2.2.9.3 POOLCHECK System Parameter	2-15
2.2.9.4 New Granularity Hint Region System Parameters	2-16
2.2.10 Using the SYSMAN Utility to Configure the I/O Subsystem	2-16
2.2.11 Symmetric Multiprocessing on AXP Systems	2-18
2.2.12 Startup Command Procedures	2-22
2.2.13 Devices on OpenVMS AXP	2-22
2.2.14 Local DSA Device Naming	2-23
2.2.15 File Name Format of Drivers Supplied by Digital on AXP	2-24
2.2.16 OpenVMS Installation Media	2-24

2.2.17	VMSINSTAL Utility	2-24
2.2.17.1	History File of VMSINSTAL Executions	2-25
2.2.17.2	Product Installation Log File	2-25
2.2.17.3	Procedure for Listing Installed Products	2-25
2.2.18	Running AUTOGEN	2-26
2.2.19	Improving the Performance of Main Images and Shareable Images	2-26
2.2.20	SYS.EXE Renamed to SYS\$BASE_IMAGE.EXE	2-26
2.2.21	Rounding-Up Algorithm for Input Quota Values	2-27
2.2.22	Terminal Fallback Facility	2-27

3 Maintenance Tasks

3.1	Maintenance Tasks That Are the Same	3-1
3.2	Maintenance Tasks That Are Different	3-3
3.2.1	DECEvent Event Management Utility	3-4
3.2.2	Comparison of Batch and Print Queuing Systems	3-4
3.2.3	System Dump Analyzer	3-7
3.2.3.1	Size of the System Dump File	3-7
3.2.3.2	Conserving Dump File Storage Space	3-8
3.2.3.3	SDA Automatically Invoked at System Startup	3-9
3.2.3.4	Using SDA CLUE Commands to Obtain and Analyze Crash Dump Information	3-9
3.2.4	Patch Utility Not Supported	3-10

4 Security Tasks

5 Performance Optimization Tasks

5.1	System Parameters: Measurement Change for Larger Page Size	5-1
5.1.1	System Parameter Units That Changed in Name Only	5-2
5.1.2	CPU-Specific System Parameter Units	5-2
5.1.3	System Parameters with Dual Values	5-3
5.2	Comparison of System Parameter Default Values	5-5
5.3	Use of Page or Pagelet Values in Utilities and Commands	5-7
5.4	Adaptive Pool Management	5-11
5.5	Installing Main Images and Shareable Images In GHRs	5-13
5.5.1	Install Utility Support	5-14
5.5.2	System Parameter Support	5-14
5.5.3	SHOW MEMORY Support	5-15
5.5.4	Loader Changes for Executive Images in GHRs	5-15
5.6	Virtual I/O Cache	5-17

6 Network Management Tasks

6.1	Network Management Tasks That Are the Same	6-1
6.2	Network Management Tasks That Are Different	6-3
6.2.1	Level 1 Routing Supported for Cluster Alias Only	6-3
6.2.2	CI and DDCCMP Lines Not Supported	6-4
6.2.3	DNS Node Name Interface Not Supported	6-4
6.2.4	VAX P.S.I. Not Supported	6-4
6.2.5	NCP Command Parameters Affected by Unsupported Features	6-5

A I/O Subsystem Configuration Commands in SYSMAN

A.1	I/O Subsystem Configuration Support in SYSMAN	A-1
	IO AUTOCONFIGURE	A-1
	IO CONNECT	A-3
	IO LOAD	A-6
	IO SET PREFIX	A-6
	IO SHOW BUS	A-7
	IO SHOW DEVICE	A-8
	IO SHOW PREFIX	A-10

B Additional Considerations

B.1	Help Message Utility	B-1
B.2	Online Documentation on Compact Disc	B-2
B.3	Unsupported DCL Commands	B-2
B.4	Password Generation	B-2
B.5	Default Editor for EDIT Command	B-2
B.6	TECO Editor	B-3
B.7	Shareable Images in the DEC C RTL for OpenVMS AXP	B-3
B.8	Run-Time Libraries	B-4
B.9	Compatibility Between the OpenVMS VAX and OpenVMS AXP Mathematics Libraries	B-4
B.10	Linker Utility Enhancements	B-4
B.10.1	New /DSF Qualifier	B-5
B.10.2	New /ATTRIBUTES Qualifier for COLLECT= Option	B-5

Index

Examples

2-1	Using ARCH_TYPE to Determine Architecture Type	2-8
2-2	Using F\$GETSYI to Display Hardware Type, Architecture Type, and Page Size	2-8

Figures

1-1	VAX Page Size, AXP Page Size, and AXP Pagelet Size	1-3
5-1	Traditional Loads and Loads into GHRs	5-16

Tables

2-1	Identical or Similar OpenVMS Setup Tasks	2-1
2-2	Comparison of VMSINSTAL and POLYCENTER Software Installation Utility	2-7
2-3	F\$GETSYI Arguments That Specify Host Architecture	2-8
2-4	Boot Flags and Their Values	2-11
2-5	Comparison of I/O Subsystem Configuration Commands	2-16
2-6	MULTIPROCESSING Values on AXP and VAX Systems	2-18
2-7	Comparison of Device Naming on OpenVMS	2-23

3-1	Identical or Similar OpenVMS Maintenance Tasks	3-1
3-2	Comparison of Batch and Print Queuing Systems	3-4
3-3	Comparison of DUMPSTYLE System Parameter Values	3-8
5-1	System Parameter Units That Changed in Name Only	5-2
5-2	CPU-Specific System Parameter Units	5-3
5-3	System Parameters with Dual Values	5-3
5-4	Comparison of System Parameter Default Values	5-6
5-5	System Parameters Associated with GHR Feature	5-15
6-1	Identical or Similar OpenVMS Network Management Tasks	6-1
6-2	NCP Command Parameters Affected by Unsupported Features	6-5
A-1	/SELECT Qualifier Examples	A-2
B-1	Run-Time Libraries Not Included in OpenVMS AXP Version 6.1	B-4

Preface

Intended Audience

This document is intended for experienced OpenVMS VAX system managers who are establishing an OpenVMS computing environment on AXP computers.

Document Structure

This document consists of six chapters, two appendixes, and an index.

Chapter 1 explains why there are differences in OpenVMS system management on AXP and VAX computers.

Chapter 2 describes how OpenVMS system management setup tasks are similar or different on AXP and VAX computers.

Chapter 3 describes how OpenVMS system management maintenance tasks are similar or different on AXP and VAX computers.

Chapter 4 describes how OpenVMS system management security tasks are similar or different on AXP and VAX computers.

Chapter 5 describes how the OpenVMS system management tasks that are designed to optimize performance are similar or different on AXP and VAX computers.

Chapter 6 describes how DECnet network management tasks are similar or different on AXP and VAX computers.

Appendix A contains reference descriptions of the I/O subsystem configuration commands that are in the System Management utility (SYSMAN).

Appendix B contains additional system management considerations that might be pertinent to your job of supporting OpenVMS general users and programmers working on AXP computers.

Associated Documents

Depending on your experience level, this manual should provide most of the basic information you will need to get started with the management of OpenVMS AXP systems. However, you should read the following two documents thoroughly:

- *OpenVMS AXP Version 6.1 Release Notes*
- *OpenVMS AXP Version 6.1 Upgrade and Installation Manual*

If your computing environment will include DECwindows Motif for OpenVMS, read:

- *DECwindows Motif Version 1.2 for OpenVMS Release Notes*
- *DECwindows Motif Version 1.2 for OpenVMS Installation Guide*

You also might want to consult the following OpenVMS manuals for related information:

- *OpenVMS Compatibility Between VAX and AXP*
- *OpenVMS DCL Dictionary*
- *VMScluster Systems for OpenVMS*
- *OpenVMS System Manager's Manual: Essentials*
- *OpenVMS System Manager's Manual: Tuning, Monitoring, and Complex Systems*
- *OpenVMS System Management Utilities Reference Manual*
- *DECnet for OpenVMS Networking Manual*
- *DECnet for OpenVMS Network Management Utilities*
- *OpenVMS Guide to System Security*
- *OpenVMS AXP System Dump Analyzer Utility Manual*
- *OpenVMS VAX System Dump Analyzer Utility Manual*

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.

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

The following conventions are also used in this manual:

Ctrl/x	A sequence such as Ctrl/x indicates that you must hold down the key labeled Ctrl while you press another key or a pointing device button.
.	Vertical ellipsis points indicate the omission of items from a code example or command format; the items are omitted because they are not important to the topic being discussed.
[]	In command format descriptions, brackets indicate optional elements. You can choose one, none, or all of the options. (Brackets are not optional, however, in the syntax of a directory name in an OpenVMS file specification or in the syntax of a substring specification in an assignment statement.)
{ }	In command format descriptions, braces surround a required choice of options; you must choose one of the options listed.
boldface text	Boldface text represents the introduction of a new term or the name of an argument, an attribute, or a reason (user action that triggers a callback). Boldface text is also used to show user input in Bookreader versions of the manual.

italic text

Italic text emphasizes important information and indicates complete titles of manuals and variables. Variables include information that varies in system messages (Internal error *number*), in command lines (/PRODUCER=*name*), and in command parameters in text (where *device-name* contains up to five alphanumeric characters).

UPPERCASE TEXT

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

-

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

numbers

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

Overview of OpenVMS AXP System Management

Most of the OpenVMS VAX Version 6.1 system management utilities, command formats, and tasks are identical in the OpenVMS AXP Version 6.1 environment. There are some differences that must be considered to set up, maintain, secure, and optimize OpenVMS AXP systems properly and to establish network connections.

Read this manual if you know about most OpenVMS VAX system management features and only need to learn what is new, identical, or different in OpenVMS AXP system management.

This manual compares system management on OpenVMS AXP Version 6.1 with:

- VMS Version 5.4
- VMS Version 5.5
- OpenVMS VAX Version 6.0
- OpenVMS VAX Version 6.1

A goal of this manual is to help you manage AXP and VAX nodes at the same time.

This chapter explains why some differences exist between OpenVMS AXP and OpenVMS VAX system management. In subsequent chapters:

- Chapter 2 compares the *setup* features and tasks.
- Chapter 3 compares the *maintenance* features and tasks.
- Chapter 4 compares the *security* features and tasks.
- Chapter 5 compares the *performance optimization* features and tasks.
- Chapter 6 compares the *network* management features and tasks.

You will find supporting system management information in the appendixes to this document. Appendix A contains reference material for the I/O subsystem configuration capabilities that have moved from the System Generation utility (SYSGEN) to the System Management utility (SYSMAN). Appendix B describes additional considerations related to your task of supporting general users and programmers on OpenVMS AXP systems.

Overview of OpenVMS AXP System Management

1.1 Reduced Number of System Management Differences

1.1 Reduced Number of System Management Differences

This release of OpenVMS AXP greatly reduces the number of differences in system management between OpenVMS AXP and OpenVMS VAX environments. Starting with OpenVMS AXP Version 6.1, a number of key features and related products are present, including:

- Fully functional VMScluster systems. The same features in VAXclusters are available in VMSclusters. VMSclusters offer the additional capability of supporting dual architectures, AXP and VAX.
- Volume Shadowing for OpenVMS.
- RMS Journaling for OpenVMS.
- The same support for multiple queue managers as in OpenVMS VAX Version 6.0 and Version 6.1.
- User-written device drivers.
- The same C2 security features as in OpenVMS VAX Version 6.0 and later releases, which are certified by the U.S. government as C2 compliant. OpenVMS AXP Version 6.1 has not been certified as C2 compliant at this time. Also, as pointed out in Chapter 4, the OpenVMS AXP C2 features do not include DECnet connection auditing.
- The POLYCENTER Software Installation utility, which is available on OpenVMS AXP Version 6.1 and OpenVMS VAX Version 6.1 systems for rapid installation or deinstallation of products and for managing information about the products installed on your systems.
- The movefile subfunction for atomic-file disk-defragmentation applications, and support of the layered software product DEC File Optimizer for OpenVMS AXP.

1.2 Why Do Some Differences Still Exist?

Why do some system management differences continue to exist in OpenVMS AXP and OpenVMS VAX environments? The following sections summarize the significant reasons for these differences. The remaining chapters in this document provide more details and will help you identify the OpenVMS AXP and OpenVMS VAX system management characteristics.

1.2.1 Different Page Size

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

Overview of OpenVMS AXP System Management

1.2 Why Do Some Differences Still Exist?

- 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. Also, on an AXP computer with 8KB pages, 16 AXP pagelets equal 1 AXP page.

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 with OpenVMS VAX users, system managers, and application programmers who are accustomed to thinking about memory values in 512-byte units. In a VMScluster, which can include certain versions of OpenVMS VAX nodes and OpenVMS AXP 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.

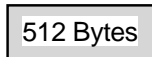
Figure 1–1 illustrates the relative sizes of a VAX page, an AXP 8KB page, and an AXP pagelet.

Figure 1–1 VAX Page Size, AXP Page Size, and AXP Pagelet Size

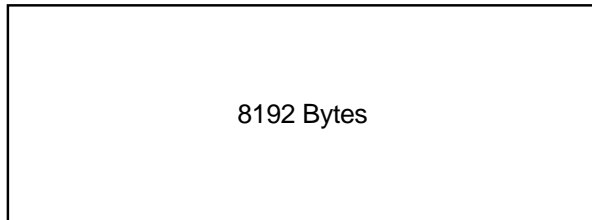
On a VAX Computer

On an AXP Computer with 8KB Pages

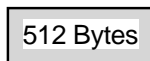
1 page:



1 page:



1 pagelet:



16 pagelets within 1 page:

512	512	512	512
512	512	512	512
512	512	512	512
512	512	512	512

ZK-5350A-GE

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

Overview of OpenVMS AXP System Management

1.2 Why Do Some Differences Still Exist?

operate with the understanding that each VAX page value and each AXP pagelet value equal a common 512-byte quantity.

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, 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 (8KB pages, 16KB pages, 32KB pages, or 64KB pages). In general, OpenVMS AXP utilities display CPU-specific page values when the data represents physical memory.

Page and pagelet units are discussed in many sections of this manual; see especially Section 2.2.21, Section 5.1, Section 5.2, and Section 5.3.

1.2.2 A New Way to Perform Standalone Backups and Other Tasks

On AXP systems, you can use a menu-driven procedure (which starts when you boot the OpenVMS AXP operating system distribution compact disc) to perform the following tasks:

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

For more detailed information about using the menu-driven procedure, see the *OpenVMS AXP Version 6.1 Upgrade and Installation Manual*.

1.2.3 DECEvent Event Management Utility

OpenVMS AXP Version 6.1 includes the DECEvent utility, which provides the interface between a system user and the system’s event log files. This allows system users to produce ASCII reports derived from system event entries. DECEvent uses the system event log file, `SYSS$ERRORLOG:ERRLOG.SYS`, as the default input file for event reporting unless another file is specified.

Note

The DECEvent utility is not available on OpenVMS VAX systems.

See Section 3.2.1 for a summary of the DECEvent utility.

1.2.4 I/O Subsystem Configuration Commands in SYSMAN

On OpenVMS VAX computers, the System Generation utility (SYSGEN) is used to modify system parameters, load device drivers, load page and swap files, and create additional page and swap files. To load device drivers on OpenVMS AXP computers, you use the System Management utility (SYSMAN) instead of SYSGEN. OpenVMS AXP SYSGEN is available for modifying system parameters,¹ loading page and swap files, and creating additional page and swap files. OpenVMS VAX 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. See Chapter 2 and Appendix A for details about SYSMAN IO commands.

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).

Overview of OpenVMS AXP System Management

1.2 Why Do Some Differences Still Exist?

1.2.5 MONITOR POOL Command Not Necessary

The DCL command MONITOR POOL that is used on VMS Version 5.5 and earlier releases is not provided on OpenVMS AXP systems or on OpenVMS VAX Version 6.0 and later releases. MONITOR POOL functions are no longer needed due to adaptive pool management features present in the operating system. See Section 5.4 for details about adaptive pool management.

1.2.6 Changes in OpenVMS File Names

The names of some command procedure files supplied by the operating system changed. For example, SYSTARTUP_V5.COM from VMS Version 5.5 and earlier releases is called SYSTARTUP_VMS.COM on OpenVMS AXP and on OpenVMS VAX Version 6.0 and later releases. Also, the VAXVMSSYS.PAR system parameter file is called ALPHAVMSSYS.PAR on OpenVMS AXP. See Chapter 2.

1.2.7 Unsupported Features in DECnet for OpenVMS AXP

Some networking features in DECnet for OpenVMS VAX (Phase IV software) are not available with DECnet for OpenVMS AXP. These unsupported features on OpenVMS AXP systems are:

- The DECnet for OpenVMS DECdns node name interface.
- VAX P.S.I.; however, a DECnet for OpenVMS AXP node can communicate with DECnet nodes that are connected to X.25 networks reachable via a DECnet/X.25 router.
- Level 2 host-based DECnet routing. Level 1 routing is supported for use only with cluster alias routers and is restricted for use on one circuit.
- DDCMP network connections.

Note

If you install the version of DECnet/OSI that is for OpenVMS AXP Version 6.1:

- DECdns client is available.
 - X.25 support is available through DEC X.25 for OpenVMS AXP. The QIO interface drivers from the P.S.I. product are included in DEC X.25 in order to provide the same interface to customer applications.
-

1.2.8 Unsupported Optional Layered Products

Most optional layered products from Digital and other vendors are supported on OpenVMS AXP systems. However, some optional layered products remain unsupported at this time. If you copy existing startup procedures from one of your OpenVMS VAX computers to an OpenVMS AXP system disk, you must comment out the calls to the startup procedures of currently unsupported optional layered products.

The *Alpha AXP Applications Catalog* lists all the available Digital optional software products and third-party applications. You can obtain a copy in the United States and Canada by calling 1-800-DIGITAL (1-800-344-4825), selecting the option for "...pre-purchased product information," and speaking with a Digital representative.

Overview of OpenVMS AXP System Management

1.2 Why Do Some Differences Still Exist?

In other locations, you can obtain the catalog from your Digital account representative or authorized reseller.

System Setup Tasks

System management setup tasks are those you perform to get the OpenVMS system installed, booted, and ready for users. This chapter:

- Identifies which OpenVMS system management setup tasks are the same on AXP and VAX computers
- Explains which OpenVMS system management setup tasks are different or new on AXP computers

2.1 Setup Tasks That Are the Same

Table 2–1 lists the OpenVMS system management setup tasks that are identical or similar on AXP and VAX.

Table 2–1 Identical or Similar OpenVMS Setup Tasks

Feature or Task	Comments
System disk directory structure	Directory structure is the same (SYSS\$SYSTEM, SYSS\$MANAGER, SYSS\$UPDATE, and so on).
Site-independent STARTUP.COM command procedure	Each OpenVMS release ships a new SYSS\$SYSTEM:STARTUP.COM procedure. Do not modify STARTUP.COM.
Site-specific startup command procedures	OpenVMS includes the following site-specific startup command procedures: SYPAG\$SWPFILES.COM, SYCONFIG.COM, SYLOGICAL.COM, and SYSECURITY.COM. The VMS SYSTARTUP_V5.COM procedure is called SYSTARTUP_VMS.COM in OpenVMS AXP and in OpenVMS VAX Version 6.0 and later releases.
VMSINSTAL procedure	Although the VMSINSTAL procedure is still available on OpenVMS AXP systems, updated layered products will use the POLYCENTER Software Installation utility instead. The POLYCENTER Software Installation utility is available for rapid installations and deinstallations, and for managing information about installed products on OpenVMS AXP Version 6.1 and OpenVMS VAX Version 6.1 systems. See Section 2.2.2 for a comparison of VMSINSTAL and the POLYCENTER Software Installation utility.
Decompressing libraries as a postinstallation task	Use @SYSS\$UPDATE:LIBDECOMP.COM as usual if you choose to decompress the system libraries (recommended).

(continued on next page)

System Setup Tasks

2.1 Setup Tasks That Are the Same

Table 2–1 (Cont.) Identical or Similar OpenVMS Setup Tasks

Feature or Task	Comments
VAXcluster system	<p data-bbox="672 331 1386 516">With this release of OpenVMS AXP, VMScluster systems include all the features available in VAXcluster systems. As with VAXcluster systems, there are limits to: the number of nodes in a VMScluster, the operating system versions that can operate together, and the layered products that can operate together. VMScluster systems offer the additional capability of supporting the AXP and VAX architectures.</p> <p data-bbox="672 527 1386 579">Essentially, all features are common between VAXcluster systems and VMScluster systems. They include:</p> <ul data-bbox="672 600 1386 1881" style="list-style-type: none"> <li data-bbox="672 600 1386 737">• 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.) <li data-bbox="672 751 1386 804">• Shared batch and print queues are accessible from any system in the VMScluster system. <li data-bbox="672 825 1386 877">• OpenVMS lock manager system services operate for all nodes in a VMScluster system. <li data-bbox="672 898 1386 1010">• 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.) <li data-bbox="672 1031 1386 1083">• AXP and VAX processors can serve TMSCP tapes to all CPUs (AXP and VAX). <li data-bbox="672 1104 1386 1178">• Process information and control services are available to application programs and system utilities on all nodes in a VMScluster system. <li data-bbox="672 1199 1386 1272">• Configuration command procedures assist in adding and removing systems and in modifying their configuration characteristics. <li data-bbox="672 1293 1386 1430">• An availability manager for VMSclusters (DECcmds), which is optionally installable, for monitoring AXP or VAX nodes in the cluster. All features are supported for AXP nodes, with an exception: the DECcmds console application cannot run on an AXP node. <li data-bbox="672 1451 1386 1503">• Booting of AXP or VAX satellite nodes can be done from an AXP server node or a VAX server node. <li data-bbox="672 1524 1386 1598">• The Show Cluster utility displays the status of all VMScluster hardware components and communication links. <li data-bbox="672 1619 1386 1692">• Standard OpenVMS system and security features work in a VMScluster system such that the entire VMScluster system operates as a single security domain. <li data-bbox="672 1713 1386 1787">• The VMScluster software balances the interconnect I/O load in VMScluster configurations that include multiple interconnects. <li data-bbox="672 1808 1386 1881">• Multiple VMScluster systems can be configured on a single or extended local area network (LAN).

(continued on next page)

System Setup Tasks

2.1 Setup Tasks That Are the Same

Table 2–1 (Cont.) Identical or Similar OpenVMS Setup Tasks

Feature or Task	Comments
	<ul style="list-style-type: none"> • The /CLUSTER qualifier to the Monitor utility operates across the entire VMScluster. • VMScluster systems support the following interconnects for AXP computers: <ul style="list-style-type: none"> – CI computer interconnect – DIGITAL Storage System Interconnect (DSSI) – Ethernet – FDDI <p>A maximum of 96 systems can be configured in a VMScluster system, the same limit as in a VAXcluster system. For configuration limits, refer to the VMScluster Software for OpenVMS AXP <i>Software Product Description</i> (SPD 42.18.xx). See <i>VMScluster Systems for OpenVMS</i> and the <i>OpenVMS AXP Version 6.1 Release Notes</i> for detailed information about VMScluster systems.</p>
Volume Shadowing for OpenVMS	<p>With this release of OpenVMS AXP, the Volume Shadowing for OpenVMS product is identical on AXP and VAX systems. Minor exceptions are noted in <i>Volume Shadowing for OpenVMS</i> and in the <i>OpenVMS AXP Version 6.1 Release Notes</i>.</p>
RMS Journaling for OpenVMS	<p>Identical, starting with OpenVMS AXP Version 6.1. Consequently, the following commands are available on OpenVMS AXP nodes:</p> <ul style="list-style-type: none"> • SET FILE/AI_JOURNAL • SET FILE/BI_JOURNAL • SET FILE/RU_ACTIVE • SET FILE/RU_FACILITY • SET FILE/RU_JOURNAL • RECOVER/RMS_FILE

(continued on next page)

System Setup Tasks

2.1 Setup Tasks That Are the Same

Table 2–1 (Cont.) Identical or Similar OpenVMS Setup Tasks

Feature or Task	Comments
User-written device drivers	<p>Similar on VAX and AXP. OpenVMS AXP Version 1.5 included 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, provided device support for customers with critical needs.</p> <p>The Step 1 driver interface has been replaced by the Step 2 driver interface in OpenVMS AXP Version 6.1. The Step 2 interface design facilitates the coding and ensures a longer life for any new device drivers. Unlike Step 1 drivers, Step 2 drivers can be written in C and can conform to the OpenVMS calling standard. Any Step 1 device driver for earlier versions of OpenVMS AXP must be replaced with Step 2 drivers on OpenVMS AXP Version 6.1. For more information about Step 2 drivers, see <i>Creating an OpenVMS AXP Step 2 Device Driver from a Step 1 Device Driver</i>.</p>
Backing up data	<p>The BACKUP command and qualifiers are the same on OpenVMS AXP. Important: Thoroughly read the <i>OpenVMS AXP Version 6.1 Release Notes</i> for the latest information about any restrictions with backup and restore operations on AXP and VAX systems.</p>
LAT startup	<p>You must start DECnet before you start LAT. As on OpenVMS VAX, always start LAT from the SYSTEM account. This account has appropriate privileges and quotas. LAT functions better when the LATACP process is running under UIC [1,4].</p> <p>You can add the following command to the SYSTARTUP_VMS.COM procedure that resides in the SYSSMANAGER directory:</p> <pre>\$ @SYS\$STARTUP:LAT\$STARTUP.COM</pre>
Local Area Transport Control Program (LATCP)	<p>Features are identical. To use LATCP, set the system parameter MAXBUF to 8192 or higher. Different systems might require different settings for MAXBUF. The BYTLM quota for accounts that use LATCP should be set accordingly in the Authorize utility.</p>
LATSYM symbiont	<p>Identical. Use LATSYM to set up LAT print queues. See the <i>OpenVMS System Manager's Manual</i> for more information.</p>
LTPAD process	<p>LTPAD provides outgoing SET HOST/LAT functionality. Service responder and outgoing connections on an AXP computer can be enabled.</p>
STARTUP SET OPTIONS and SHUTDOWN NODE commands in SYSMAN	<p>Identical on OpenVMS AXP, starting with Version 6.1.</p>

(continued on next page)

System Setup Tasks

2.1 Setup Tasks That Are the Same

Table 2–1 (Cont.) Identical or Similar OpenVMS Setup Tasks

Feature or Task	Comments
Digital's InfoServer, which includes the Local Area Disk Control Program (LADCP), LASTport network transport control program, LASTport /Disk protocol, LASTport /Tape protocol	Identical.
Security audit log file name	The same name on OpenVMS AXP Version 6.1 and OpenVMS VAX Version 6.0 and later releases: SECURITY.AUDIT\$JOURNAL. Called SECURITY_AUDIT.AUDIT\$JOURNAL on earlier OpenVMS AXP releases and on VAX VMS Version 5.5 and earlier releases. Resides in SYSS\$COMMON:[SYSMGR] in both cases.
DECdtm and its two-phase commit protocol	Identical.
VMSTAILOR and DECW\$TAILOR utilities	Identical.
Authorize utility	Identical.
OPCOM	Identical.
Terminal Fallback Facility (TFF)	Similar function but with some differences. See Section 2.2.22 for details.
User Environment Test Package (UETP)	Identical.

2.2 Setup Tasks That Are Different

This section describes the OpenVMS system management setup tasks that are different or new on AXP. Differences are in the following areas:

- On OpenVMS AXP systems, there is a new way to back up a system disk (as an alternative to Standalone BACKUP, which is used on OpenVMS VAX systems). See Section 2.2.1.
- Installations, depending on whether the product you are installing uses the VMSINSTALL procedure or the new POLYCENTER Software Installation utility. See Section 2.2.2.
- Planning for and managing common object and image file extensions. See Section 2.2.3.
- The format of the BOOT command on AXP systems and the boot flags are different. See Section 2.2.4.
- The CONSCOPY.COM command procedure is not supplied with the OpenVMS AXP kit. See Section 2.2.5.
- Use of the License Management Facility (LMF). See Section 2.2.6.
- One of the two Product Authorization Keys (PAKs) for DECnet for OpenVMS AXP has a different name from one of the PAK names on VAX systems. See Section 2.2.7.

System Setup Tasks

2.2 Setup Tasks That Are Different

- The PAK name for VMSclusters is different from the PAK name for VAXclusters. See Section 2.2.8.
- System Generation utility (SYSGEN) and its parameters. See Section 2.2.9.
- I/O subsystem configuration commands, controlled in SYSGEN on OpenVMS VAX, are in the OpenVMS AXP System Management utility (SYSMAN). See Section 2.2.10.
- Symmetric multiprocessing (SMP). See Section 2.2.11.
- Startup command procedure changes because of relocation of I/O subsystem configuration functions from SYSGEN to SYSMAN. See Section 2.2.12.
- Hardware devices on AXP computers. See Section 2.2.13.
- DIGITAL Storage Architecture (DSA) device naming. See Section 2.2.14.
- The file name format of drivers supplied by Digital on AXP systems. See Section 2.2.15.
- OpenVMS installation media contains binaries and documentation. See Section 2.2.16.
- VMSINSTAL utility. See Section 2.2.17.
- Running the AUTOGEN procedure. See Section 2.2.18.
- Improving the performance of main images and shareable images by using a feature called granularity hint regions. See Section 2.2.19.
- SYS.EXE loadable executive image renamed to SYS\$BASE_IMAGE.EXE. See Section 2.2.20.
- The rounding-up algorithm used when you input values for quotas that are in quantities of pagelets. See Section 2.2.21.
- Terminal Fallback Facility (TFF). See Section 2.2.22.

2.2.1 New Way to Back Up System Disks

On AXP systems, you can use a menu-driven procedure (which starts when you boot the OpenVMS AXP operating system distribution compact disc) to perform the following tasks:

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

For more detailed information about using the menu-driven procedure, see the *OpenVMS AXP Version 6.1 Upgrade and Installation Manual*. See Section 2.2.2 for a summary of the POLYCENTER Software Installation utility.

2.2.2 VMSINSTAL and the POLYCENTER Software Installation Utility

Although the VMSINSTAL procedure is still available on OpenVMS AXP systems, updated layered products will use the POLYCENTER Software Installation utility instead. The POLYCENTER Software Installation utility is available for rapid installations and deinstallations, and for managing information about installed products on OpenVMS AXP Version 6.1 and OpenVMS VAX Version 6.1 systems. OpenVMS AXP Version 1.5 and earlier systems, and OpenVMS VAX Version 6.0 and earlier systems, all use VMSINSTAL for installations.

System Setup Tasks

2.2 Setup Tasks That Are Different

Table 2–2 compares the VMSINSTAL procedure and the POLYCENTER Software Installation utility.

Table 2–2 Comparison of VMSINSTAL and POLYCENTER Software Installation Utility

VMSINSTAL Procedure	POLYCENTER Software Installation Utility
Begins an installation before you make configuration choices.	You can perform an installation by choosing your software configuration and then installing the product.
When you press Ctrl/Y, the procedure leaves your system in the same state it was in prior to the start of the installation.	If you press Ctrl/Y (or click on the Cancel button) while the POLYCENTER Software Installation utility is installing a product, you must perform a remove operation to clean up any files created by the partial installation.
Prompts you as to whether you want to back up your system disk.	Does not ask you whether you want to back up your system disk before you begin an installation.

See the following documents for detailed information about the POLYCENTER Software Installation utility:

- *POLYCENTER Software Installation Utility User's Guide*
- *POLYCENTER Software Installation Utility Developer's Guide*

2.2.3 Planning for and Managing Common Object and Image File Extensions

File extensions on OpenVMS VAX computers are identical on OpenVMS AXP computers, including .OBJ for object files and .EXE for executable files. It is important that you plan for and track the location of the following files, especially in VMSclusters that include AXP and VAX systems using common disks:

- Native, VAX specific .OBJ and .EXE files (to be linked or executed on OpenVMS VAX nodes only).
- Native, AXP specific .OBJ and .EXE files (to be linked or executed on OpenVMS AXP nodes only).
- Translated VAX .EXE images (to be executed on OpenVMS AXP nodes only). An OpenVMS VAX image named *file.EXE* becomes *file_TV.EXE* when translated.

Determining the Host Architecture

Use the F\$GETSYI lexical function from within your command procedure, or the \$GETSYI system service or the LIB\$GETSYI RTL routine from within your program, to determine whether the procedure or program is running on an OpenVMS VAX system or on an OpenVMS AXP system.

Example 2–1 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*.

System Setup Tasks

2.2 Setup Tasks That Are Different

Example 2–1 Using ARCH_TYPE 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
```

Once the architecture type is determined, the command procedure or program can branch to the appropriate code for architecture-specific processing.

The ARCH_TYPE argument and other useful F\$GETSYI arguments are summarized in Table 2–3.

Table 2–3 F\$GETSYI Arguments That Specify Host Architecture

Argument	Usage
ARCH_TYPE	Returns “1” on a VAX; returns “2” on an AXP
ARCH_NAME	Returns “VAX” on a VAX; returns “Alpha” on an AXP
PAGE_SIZE	Returns “512” on a VAX; returns “8192” on an AXP

Example 2–2 shows a simple command procedure that uses F\$GETSYI to display several architecture-specific values.

Example 2–2 Using F\$GETSYI to Display Hardware Type, Architecture Type, and Page Size

```
$ ! File name: SHOW_ARCH.COM
$ !
$ ! Simple command procedure to display node hardware type,
$ ! architecture type, page size, and other basic information.
$ !
$ say = "write sys$output"
$ say " "
$ say "OpenVMS process with PID " + "'f$getjpi(", "PID)'"
$ say " running at " + "'f$time()'" + "."
$ say " "
$ say "Executing on a " + "'f$getsyi("HW_NAME)'"
$ say " named " + "'f$getsyi("NODENAME)'" + "."
$ say " "
$ say "Architecture type is " + "'f$getsyi("ARCH_TYPE)'"
$ say " and architecture name is " + "'f$getsyi("ARCH_NAME)'" + "."
$ say " "
$ say "Page size is " + "'f$getsyi("PAGE_SIZE)'" + " bytes."
$ !
$ exit
```

System Setup Tasks

2.2 Setup Tasks That Are Different

On an OpenVMS VAX Version 6.1 node, output from the procedure is similar to the following display:

```
OpenVMS process with PID 3FE00B0E
  running at 16-MAY-1994 04:23:07.92.
```

```
Executing on a VAX 6000-620
  named NODEXX.
```

```
Architecture type is 1
  and architecture name is VAX.
```

```
Page size is 512 bytes.
```

On an OpenVMS AXP Version 6.1 node, output from the procedure is similar to the following display:

```
OpenVMS process with PID 2FC00126
  running at 16-MAY-1994 04:23:59.37.
```

```
Executing on a DEC 4000 Model 610
  named SAMPLE.
```

```
Architecture type is 2
  and architecture name is Alpha.
```

```
Page size is 8192 bytes.
```

Note

For the F\$GETSYI lexical function, the PAGE_SIZE, ARCH_NAME, and ARCH_TYPE arguments do not exist on VMS systems predating Version 5.5. On VMS Version 5.4 and earlier systems, you can use the F\$GETSYI("CPU") lexical function.

Output of Analyze Utility Commands

The display created by ANALYZE/OBJECT and ANALYZE/IMAGE commands on an OpenVMS AXP node identifies the architecture type of an .OBJ or .EXE file. The OpenVMS AXP command ANALYZE/OBJECT works with AXP or VAX object files. Similarly, the OpenVMS AXP command ANALYZE/IMAGE works with AXP or VAX image files. The OpenVMS VAX ANALYZE/OBJECT and ANALYZE/IMAGE commands do not have this capability.

- When you enter an ANALYZE/IMAGE command on an OpenVMS AXP node and the image being analyzed is an OpenVMS VAX image file, the following text is included on the first page of the displayed report:

```
This is an OpenVMS VAX image file
```

- When you enter an ANALYZE/OBJECT command on an OpenVMS AXP node and the object being analyzed is an OpenVMS VAX object file, the following text is included on the first page of the displayed report:

```
This is an OpenVMS VAX object file
```

- When you enter an ANALYZE/IMAGE command on an OpenVMS AXP node and the image being analyzed is an OpenVMS AXP image file, the following text is included on the first page of the displayed report:

```
This is an OpenVMS Alpha image file
```

System Setup Tasks

2.2 Setup Tasks That Are Different

- When you enter an ANALYZE/OBJECT command on an OpenVMS AXP node and the object being analyzed is an OpenVMS AXP object file, the following text is included on the first page of the displayed report:

```
This is an OpenVMS Alpha object file
```

On an OpenVMS VAX node, the LINK and RUN commands return error messages if the file that users are attempting to link or run was created by an OpenVMS AXP compiler or linker. For example:

```
$ ! On an OpenVMS VAX node
$ RUN SALARY_REPORT.EXE      ! An OpenVMS AXP image

%DCL-W-ACTIMAGE, error activating image SALARY_REPORT.EXE
-CLI-E-IMGNAME, image file _$11$DUA20:[SMITH.WORK]SALARY_REPORT.EXE;1
-IMGACT-F-IMG_SIZ, image header descriptor length is invalid
```

An error message is displayed when you attempt to execute a VAX image on an OpenVMS AXP node. For example:

```
$ ! On an OpenVMS AXP node
$ RUN PAYROLL.EXE          ! An OpenVMS VAX image

%DCL-W-ACTIMAGE, error activating image PAYROLL
-CLI-E-IMGNAME, image file DUA6:[SMITH.APPL]PAYROLL.EXE;7
-IMGACT-F-NOTNATIVE, image is not an OpenVMS Alpha image
```

2.2.4 BOOT Console Command

The AXP console software attempts to locate, load, and transfer the primary bootstrap program from the boot devices specified in the BOOT console command. The BOOT command format on AXP systems is:

```
BOOT [[-FLAGS system_root,boot_flags] [device_list]]
```

The `-FLAGS` qualifier indicates that the next two comma-separated strings are the `system_root` and `boot_flags` parameters. Console software passes both parameters to Alpha primary bootstrap (APB) without interpretation as an ASCII string like 0,0 in the environment variable `BOOTED_OSFLAGS`.

The `system_root` parameter specifies the hexadecimal number of the root directory on the system disk device in which the bootstrap files and bootstrap programs reside. A root directory is a top-level directory whose name is in the form `SYSnn`, where `nn` is the number specified by `system_root`.

The `boot_flags` parameter specifies the hexadecimal representation of the sum of the desired boot flags. Table 2-4 lists possible boot flags and their values.

The `device_list` parameter is a list of device names, delimited by commas, from which the console must attempt to boot. A device name in `device_list` does not necessarily correspond to the OpenVMS device name for a given device. In fact, console software translates the device name to a path name before it attempts the bootstrap. The path name enables the console to locate the boot device through intervening adapters, buses, and widgets (for example, a controller). The path name specification and the algorithm that translates the device name to a path name are system specific.

System Setup Tasks

2.2 Setup Tasks That Are Different

Table 2–4 Boot Flags and Their Values

Hexadecimal Value	Name	Meaning If Set
1	CONV	Bootstrap conversationally; that is, allow the console operator to modify system parameters in SYSBOOT.
2	DEBUG	Map XDELTA to running system.
4	INIBPT	Stop at initial system breakpoint.
8	DIAG	Perform diagnostic bootstrap.
10	BOOBPT	Stop at bootstrap breakpoints.
20	NOHEADER	Secondary bootstrap image contains no header.
40	NOTEST	Inhibit memory test.
80	SOLICIT	Prompt for the name of the secondary bootstrap file.
100	HALT	Halt before secondary bootstrap.
2000	CRDFAIL	Mark corrected read data error pages bad.
10000	DBG_INIT	Enable verbose mode in APB, SYSBOOT, and EXEC_INIT.
20000	USER_MSGS	Enable descriptive mode, presenting a subset of the verbose mode seen when DBG_INIT is enabled. See Section 2.2.4.1 for more information.

In response to the BOOT console command, console software attempts to boot from devices in the boot device list, starting with the first one. As it attempts to boot from a specific device, console software initializes the BOOTED_DEV environment variable with the path name of that device. If an attempt to boot from a specific device fails, console software attempts to boot from the next device in the list. If all attempts fail, console software prints an error message on the console and enters the halt state to await operator action.

Later, APB uses the value in BOOTED_DEV to determine the boot device.

2.2.4.1 DBG_INIT and USER_MSGS Boot Flags

When the DBG_INIT boot flag (bit 16) is set, many informational messages are displayed during booting. This bit normally is used during testing but could be useful for any problems with booting the computer. Bits <63:48> contain the SYS*n* root from which you are booting.

OpenVMS AXP includes a new flag, USER_MSGS, that enables descriptive booting. This flag is bit 17. Set the USER_MSGS boot flag the same way you set other boot flags.

When the USER_MSGS flag is set, messages that describe the different phases of booting are displayed. These messages guide the user through the major booting phases and are a subset of the messages displayed in verbose mode when the bit 16 DBG_INIT flag is set. The USER_MSGS flag suppresses all the test and debug messages that are displayed when bit 16 is set. Error messages are always enabled and displayed as needed.

System Setup Tasks

2.2 Setup Tasks That Are Different

The following display shows a partial boot session with the USER_MSGS flag set:

```
INIT-S-CPU...
AUDIT_CHECKSUM_GOOD
AUDIT_LOAD_BEGINS
AUDIT_LOAD_DONE
%APB-I-APBVER, Alpha AXP Primary Bootstrap, Version X59S
%APB-I-BOOTDEV, Determining boot device type
%APB-I-BOOTDRIV, Selecting boot driver
%APB-I-BOOTFILE, Selecting boot file
%APB-I-BOOTVOL, Mounting boot volume
%APB-I-OPBOOTFILE, Opening boot file
%APB-I-LOADFILE, Loading [SYS0.SYSCOMMON.SYSEXE]SYSBOOT.EXE;1
%APB-I-SECBOOT, Transferring to secondary bootstrap
```

In comparison, the following display shows a partial boot session with the DBG_INIT flag set. Notice that many more messages are displayed.

```
INIT-S-CPU...
AUDIT_CHECKSUM_GOOD
AUDIT_LOAD_BEGINS
AUDIT_LOAD_DONE
%APB-I-APBVER, Alpha AXP Primary Bootstrap, Version X59S
Initializing TIMEDWAIT constants...
Initializing XDELTA...
Initial breakpoint not taken...
%APB-I-BOOTDEV, Determining boot device type
Initializing the system root specification...
%APB-I-BOOTDRIV, Selecting boot driver
%APB-I-BOOTFILE, Selecting boot file
%APB-I-BOOTVOL, Mounting boot volume
Boot QIO: VA = 20084000 LEN = 00000024 LBN = 00000000 FUNC = 00000032
Boot QIO: VA = 00000000 LEN = 00000000 LBN = 00000000 FUNC = 00000008
Boot QIO: VA = 20084000 LEN = 00000012 LBN = 00000000 FUNC = 00000027
Boot QIO: VA = 20084000 LEN = 00000008 LBN = 00000000 FUNC = 00000029
Boot QIO: VA = 20086000 LEN = 00000200 LBN = 00000001 FUNC = 0000000C
Boot QIO: VA = 20086200 LEN = 00000200 LBN = 000EE962 FUNC = 0000000C
Boot QIO: VA = 2005DD38 LEN = 00000200 LBN = 000EE965 FUNC = 0000000C
Boot QIO: VA = 20088000 LEN = 00001200 LBN = 00000006 FUNC = 0000000C
%APB-I-OPBOOTFILE, Opening boot file
Boot QIO: VA = 20098000 LEN = 00000200 LBN = 000EEBFE FUNC = 0000000C
Boot QIO: VA = 20089200 LEN = 00000200 LBN = 0000001B FUNC = 0000000C
Boot QIO: VA = 20098000 LEN = 00000200 LBN = 000EEC08 FUNC = 0000000C
Boot QIO: VA = 20089400 LEN = 00000200 LBN = 0013307D FUNC = 0000000C
Boot QIO: VA = 20098000 LEN = 00000200 LBN = 000EE96B FUNC = 0000000C
Boot QIO: VA = 20089600 LEN = 00000200 LBN = 00000027 FUNC = 0000000C
Boot QIO: VA = 20098000 LEN = 00000200 LBN = 000EE975 FUNC = 0000000C
Boot QIO: VA = 20089800 LEN = 00001600 LBN = 000F2B6E FUNC = 0000000C
Boot QIO: VA = 20098000 LEN = 00000200 LBN = 000EE9DB FUNC = 0000000C
%APB-I-LOADFILE, Loading [SYS0.SYSCOMMON.SYSEXE]SYSBOOT.EXE;1
Boot QIO: VA = 2009A000 LEN = 00000200 LBN = 00111993 FUNC = 0000000C
Boot QIO: VA = 00000000 LEN = 00050200 LBN = 00111995 FUNC = 0000000C
%APB-I-SECBOOT, Transferring to secondary bootstrap
```

2.2.5 CONSCOPY.COM Command Procedure Not Available

The OpenVMS VAX kit provides the CONSCOPY.COM command procedure, which you can use to create a backup copy of the original console volume. The OpenVMS VAX installation supplies the procedure in SYSSUPDATE. The CONSCOPY.COM procedure does not exist for OpenVMS AXP computers as the AXP consoles exist in read-only memory and not on disks.

2.2.6 Use of the License Management Facility (LMF)

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

LMF Caution for OpenVMS VAX Systems Predating Version 6.1

Note

The following cautionary text is for systems using a common LDB, and involves VAX nodes that are running OpenVMS VAX Version 6.0 and earlier releases. OpenVMS VAX Version 6.1 fixes the limitations noted in the following discussion.

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

Because the License Management Facility (LMF) for OpenVMS AXP is capable of handling all types of PAKs, including those for OpenVMS VAX, Digital recommends that you perform your LDB tasks using the OpenVMS AXP LMF.

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

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

Avoid using the following LICENSE commands from an OpenVMS VAX system on a PAK containing the ALPHA option:

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

By default, all OpenVMS AXP PAKs look disabled to an OpenVMS VAX Version 6.0 or earlier system. Never use the DELETE/STATUS=DISABLED command from an OpenVMS VAX system on an LDB that contains OpenVMS AXP PAKs. If you do, all OpenVMS AXP PAKs will be deleted.

With the exception of the DELETE/STATUS=DISABLED command, if you inadvertently use one of the LICENSE commands listed previously on an OpenVMS AXP PAK while using an OpenVMS VAX Version 6.0 or earlier system, the PAK and the database probably will not be affected adversely. Repeat the command using LMF running on an OpenVMS AXP system; the PAK should return to a valid state.

System Setup Tasks

2.2 Setup Tasks That Are Different

If you fail to repeat the command using LMF on an OpenVMS AXP system, the OpenVMS AXP system will be mostly unaffected. At worst, an OpenVMS AXP PAK that you intended to disable will remain enabled. Only OpenVMS AXP LMF can disable an OpenVMS AXP PAK.

However, if you attempt to use any of the commands listed previously on a PAK located in an LDB that is shared with an OpenVMS VAX Version 6.0 or earlier system, the following serious problems may result:

- Because OpenVMS AXP PAKs look disabled to an OpenVMS VAX Version 6.0 or earlier system, they are normally ignored at load time by OpenVMS VAX systems. However, if one of the commands listed previously is entered from an OpenVMS VAX system and the PAK information is not set to a valid state by an OpenVMS AXP system, the OpenVMS VAX Version 6.0 or earlier system may attempt to load the OpenVMS AXP PAK. Because the OpenVMS VAX system will be unable to load the PAK, the OpenVMS VAX LMF will report an error.
- Even if a valid OpenVMS VAX PAK for the affected product is in the LDB, it might not load. In this case, system users may be denied access to the product.

If the PAK cannot be restored to a valid state because all OpenVMS AXP systems are inaccessible for any reason, use your OpenVMS VAX system to disable the OpenVMS AXP PAK. This prevents your VAX system from attempting to load the OpenVMS AXP PAK.

As noted previously, the LMF that is part of OpenVMS VAX Version 6.1 removes these command restrictions.

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

2.2.7 PAK Name Difference Using DECnet for OpenVMS AXP

The DECnet cluster alias feature is available on OpenVMS AXP. Note, however, that the PAK name enabling cluster alias routing support on OpenVMS AXP (DVNETEXT) is different from the PAK name enabling cluster alias routing support on OpenVMS VAX (DVNETRTG). The functions supported with the DVNETEXT license differ from the VAX DVNETRTG license. DVNETEXT is supported only to enable level 1 routing on AXP nodes acting as routers for a cluster alias.

Routing between multiple circuits is not supported. Level 2 routing is not supported on DECnet for OpenVMS AXP nodes.

The PAK name for the end node license (DVNETEND) is the same on AXP and VAX systems.

See Chapter 6 for more information about DECnet for OpenVMS AXP.

2.2.8 PAK Name Difference for VMSclusters and VAXclusters

The PAK name for VMSclusters is VMSCLUSTER. The PAK name for VAXclusters is VAXCLUSTER.

2.2.9 SYSGEN Utility and System Parameters

The OpenVMS AXP System Generation utility (SYSGEN) is available for examining and modifying system parameters on the active system and for examining and modifying the system parameter file ALPHAVMSSYS.PAR.¹ Those functions are similar to the OpenVMS VAX SYSGEN. However, OpenVMS AXP SYSGEN and OpenVMS VAX SYSGEN differ in the following ways:

- OpenVMS AXP includes several new and modified system parameters. Some of the system parameter changes are because of new features. Other changes are due to the larger page sizes of AXP computers. See Section 2.2.9.1 through Section 2.2.9.4 for information about the new system parameters; also see Chapter 5 for information about changes to system parameters.
- On OpenVMS AXP, I/O subsystem configuration capabilities have been removed from SYSGEN. The System Management utility (SYSMAN) provides this functionality on OpenVMS AXP.

Refer to Section 2.2.10 and to Appendix A for more information about the SYSMAN I/O subsystem configuration commands.

2.2.9.1 MULTIPROCESSING System Parameter

The MULTIPROCESSING system parameter controls loading of the OpenVMS system synchronization image, which is used to support symmetric multiprocessing (SMP) options on supported AXP and VAX computers.

On OpenVMS AXP systems, and on OpenVMS VAX Version 6.0 and later releases, the MULTIPROCESSING system parameter has a new value (4). When MULTIPROCESSING is set to 4, OpenVMS always loads the streamlined multiprocessing synchronization image, regardless of system configuration or CPU availability.

See Section 2.2.11 for more information about SMP.

2.2.9.2 PHYSICAL_MEMORY System Parameter

OpenVMS AXP does not have the PHYSICALPAGES system parameter. Use the system parameter PHYSICAL_MEMORY instead of PHYSICALPAGES. If you want to reduce the amount of physical memory available for use, change the PHYSICAL_MEMORY parameter. The default setting for the PHYSICAL_MEMORY parameter is -1 (unlimited).

2.2.9.3 POOLCHECK System Parameter

The adaptive pool management feature described in Section 5.4 makes use of the POOLCHECK system parameter. The feature maintains usage statistics and extends detection of pool corruption.

Two versions of the SYSTEM_PRIMITIVES executive image are provided that give you a boot-time choice of either a minimal pool-code version or a pool-code version that features statistics and corruption detection:

- POOLCHECK zero (default value)
SYSTEM_PRIMITIVES_MIN.EXE is loaded.
- POOLCHECK nonzero
SYSTEM_PRIMITIVES.EXE, pool checking, and monitoring version are loaded.

¹ The file name VAXVMSSYS.PAR is used on OpenVMS VAX systems.

System Setup Tasks

2.2 Setup Tasks That Are Different

These features are available on systems running OpenVMS AXP or OpenVMS VAX Version 6.0 and later releases. The features are not available on systems running VAX VMS Version 5.5 and earlier releases.

See Section 5.4 for more information.

2.2.9.4 New Granularity Hint Region System Parameters

Five system parameters are associated with the granularity hint regions (GHR) feature described in Section 5.5. Refer to Section 2.2.19 and Section 5.5 for more information.

2.2.10 Using the SYSMAN Utility to Configure the I/O Subsystem

Use the System Management utility (SYSMAN) on OpenVMS AXP computers to connect devices, load I/O device drivers, and debug device drivers. These functions are provided by SYSGEN on OpenVMS VAX computers.

Enter the following command to invoke SYSMAN:

```
$ RUN SYS$SYSTEM:SYSMAN
SYSMAN>
```

Appendix A contains complete format descriptions for the IO AUTOCONFIGURE, IO CONNECT, IO LOAD, IO SET PREFIX, IO SHOW BUS, IO SHOW DEVICE, and IO SHOW PREFIX commands. Table 2–5 compares the I/O subsystem configuration commands on OpenVMS AXP and OpenVMS VAX.

Table 2–5 Comparison of I/O Subsystem Configuration Commands

OpenVMS VAX SYSGEN Command	OpenVMS AXP SYSMAN Command ¹
AUTOCONFIGURE <i>adapter-spec</i> or AUTOCONFIGURE ALL.	The default for IO AUTOCONFIGURE is all devices. There is no parameter to the IO AUTOCONFIGURE command. The /SELECT and /EXCLUDE qualifiers are not mutually exclusive, as they are on OpenVMS VAX. Both qualifiers can be specified on the command line.
CONFIGURE.	Used on VAX for Q-bus and UNIBUS, which are not supported on OpenVMS AXP.
CONNECT/ADAPTER requires CMKRNL privilege only.	IO CONNECT requires CMKRNL and SYSLCK privileges.
CONNECT/ADAPTER offers the /ADPUNIT qualifier.	No equivalent.
CONNECT/ADAPTER offers the /CSR_OFFSET qualifier.	Use IO CONNECT/ADAPTER/CSR. Note: CSR is the control and status register.
CONNECT/ADAPTER offers the /DRIVERNAME (no underscore) qualifier.	IO CONNECT offers the /DRIVER_NAME qualifier.
No equivalent.	IO CONNECT offers the /LOG=(ALL,CRB,DDB,DPT,IDB,SC,UCB) qualifier and options.
CONNECT/ADAPTER offers the /MAXUNITS (no underscore) qualifier.	IO CONNECT offers the /MAX_UNITS qualifier.

¹All I/O subsystem configuration commands on OpenVMS AXP are preceded by "IO".

(continued on next page)

System Setup Tasks

2.2 Setup Tasks That Are Different

Table 2–5 (Cont.) Comparison of I/O Subsystem Configuration Commands

OpenVMS VAX SYSGEN Command	OpenVMS AXP SYSMAN Command ¹
No equivalent.	IO CONNECT offers the /NUM_UNITS qualifier.
CONNECT/ADAPTER offers the /NUMVEC (no underscore) qualifier.	IO CONNECT offers the /NUM_VEC qualifier.
CONNECT/ADAPTER uses the /SYSIDHIGH and /SYSIDLOW qualifiers.	IO CONNECT provides the /SYS_ID qualifier to indicate the SCS system ID of the remote system to which the device is to be connected.
CONNECT/ADAPTER provides the /VECTOR_OFFSET qualifier to specify the offset <i>from</i> the interrupt vector address of the multiple device board <i>to</i> the interrupt vector address for the specific device being connected.	No equivalent.
No equivalent.	IO CONNECT provides the /VECTOR_SPACING qualifier.
CONNECT CONSOLE.	OpenVMS AXP does not require this command.
LOAD requires CMKRNL privilege.	IO LOAD requires CMKRNL and SYSLCK privileges. Also, IO LOAD provides the /LOG=(ALL,DPT) qualifier to display information about drivers that have been loaded.
RELOAD.	Not supported.
No equivalent.	IO SET PREFIX sets the prefix list used to manufacture the IOGEN Configuration Building Module (ICBM) names.
SHOW/ADAPTER.	Use IO SHOW BUS, which lists all the buses, node numbers, bus names, TR numbers, and base CSR addresses.
SHOW/CONFIGURATION.	Used on VAX for Q-bus and UNIBUS, which are not supported. Use IO SHOW BUS.
SHOW/DEVICE displays full information about the device drivers loaded into the system, including the start and end address of each device driver.	The command is IO SHOW DEVICE. Start and end address information is not shown.
SHOW/DRIVER displays the start and end addresses of device drivers loaded into the system.	The command is IO SHOW DRIVER. It displays the loaded drivers but does not display the start and end addresses because drivers may be loaded into granularity hint regions.
No equivalent.	IO SHOW PREFIX displays the current prefix list used in the manufacture of ICBM names.
SHOW/UNIBUS.	No equivalent; UNIBUS devices are not supported on AXP processors.

¹All I/O subsystem configuration commands on OpenVMS AXP are preceded by "IO".

First, you should familiarize yourself with the differences between the I/O subsystem configuration commands in OpenVMS VAX SYSGEN and OpenVMS AXP SYSMAN. Next, change the DCL procedures (if you copied any over from the VAX to the AXP system) that include commands such as:

System Setup Tasks

2.2 Setup Tasks That Are Different

```
$ SYSGEN ::= $SYS$SYSTEM:SYSGEN
$ SYSGEN io-subsystem-configuration-command
```

to:

```
$ SYSMAN ::= $SYS$SYSTEM:SYSMAN
$ SYSMAN IO io-subsystem-configuration-command
```

Look for differences in the command parameters and qualifiers, as noted in Table 2-5.

Note

For OpenVMS AXP, SYSMAN IO AUTOCONFIGURE occurs automatically at startup.

2.2.11 Symmetric Multiprocessing on AXP Systems

Symmetric multiprocessing (SMP) is supported on selected OpenVMS AXP systems. Refer to the OpenVMS AXP *Software Product Description* (SPD 41.87.xx) for the most up-to-date information about supported SMP configurations.

On the supported AXP systems, SMP is enabled automatically by the console firmware as long as there are multiple CPUs and the environment variable *cpu_enabled* is set either to *ff hexadecimal* or to the mask of available CPUs. (Each bit corresponds to a CPU. For example, bit 0 corresponds to CPU 0, and so forth.)

SMP also is managed on AXP and VAX systems by using the MULTIPROCESSING system parameter. MULTIPROCESSING controls the loading of the system synchronization image. The system parameter's values of 0, 1, 2, 3, and 4 have *nearly* equivalent functions on AXP systems and on VAX systems running OpenVMS VAX Version 6.0 and later releases. Table 2-6 summarizes the functions of the five MULTIPROCESSING values.

Table 2-6 MULTIPROCESSING Values on AXP and VAX Systems

Value	Function
0	Load the uniprocessing synchronization image.
1	Load the full-checking multiprocessing synchronization image if the CPU type is capable of SMP and two or more CPUs are present on the system.
2	Always load the full-checking version, regardless of the system configuration or CPU availability.
3	On VAX, load the SPC SYSTEM_SYNCHRONIZATION image. On AXP, load the streamlined multiprocessing synchronization image. In either case, the load occurs only if the CPU type is capable of SMP and two or more CPUs are present on the system.
4	On VAX, always load the regular streamlined image. On an OpenVMS AXP system, always load the streamlined multiprocessing synchronization image. In either case, the load occurs regardless of system configuration or CPU availability.

When the full-checking multiprocessing synchronization image is loaded, OpenVMS performs software sanity checks on the node's CPUs; also, OpenVMS provides a full history of CPU information in the event of a system failure. OpenVMS stores a program counter (PC) history in the spinlock (SPL) structures used to synchronize system activity. When the system fails, that information

System Setup Tasks

2.2 Setup Tasks That Are Different

is accessible by using the SDA command SHOW SPINLOCK. The information displayed includes the PCs of the last 16 acquisitions and releases of the spin locks.

The performance of an SMP node running the full-checking image is slower compared with a node running the streamlined image. However, it is easier to debug failures on SMP nodes (if you are writing privileged code) when the full-checking image is enabled. The streamlined image is designed for faster performance, with a trade-off of less extensive debug support following a system failure.

In addition to MULTIPROCESSING, the following system parameters control the behavior of an SMP system. These parameters have equivalent functions on AXP and VAX multiprocessing systems.

- SMP_CPUS system parameter

SMP_CPUS identifies which secondary processors, if available, are to be booted into the multiprocessing system at boot time. SMP_CPUS is a 32-bit mask; if a bit in the mask is set, the processor with the corresponding CPU ID is booted into the multiprocessing system (if it is available). For example, if you want to boot only the CPUs with CPU IDs 0 and 1, specify the value 3 (both bits are on). The default value of SMP_CPUS, -1, boots all available CPUs into the multiprocessing system.

Although a bit in the mask corresponds to the primary processor's CPU ID, the primary processor is always booted. That is, if the mask is set to 0, the primary CPU will still boot. Any available secondary processors will not be booted into the multiprocessing system.

The SMP_CPUS system parameter is ignored if the MULTIPROCESSING parameter is set to 0.

- SMP_SPINWAIT system parameter

SMP_SPINWAIT establishes, in 10-microsecond intervals, the amount of time a CPU normally waits for access to a shared resource. This process is called **spinwaiting**. A timeout causes a CPUSPINWAIT bugcheck. For SMP_SPINWAIT, the default value of 100,000 10-microsecond intervals (1 second) is usually adequate.

- SMP_LNGSPINWAIT system parameter

Certain shared resources in a multiprocessing system take longer to become available than allowed for by the SMP_SPINWAIT parameter. The SMP_LNGSPINWAIT parameter establishes, in 10-microsecond intervals, the amount of time a CPU in an SMP system waits for these resources. A timeout causes a CPUSPINWAIT bugcheck. For SMP_LNGSPINWAIT, the default value of 3,000,000 10-microsecond intervals (30 seconds) is usually adequate.

- SMP_SANITY_CNT system parameter

SMP_SANITY_CNT establishes, in 10-millisecond clock ticks, the timeout interval for each CPU in a multiprocessing system. Each CPU in an SMP system monitors the sanity timer of one other CPU in the configuration to detect hardware or software failures. If allowed to go undetected, these failures could cause the system to hang. A timeout causes a CPUSANITY bugcheck. For SMP_SANITY_CNT, the default value of 300 10-millisecond intervals (3 seconds) is usually adequate.

System Setup Tasks

2.2 Setup Tasks That Are Different

The SHOW CPU command displays information about the status, characteristics, and capabilities of the processors active in and available to an OpenVMS multiprocessing system. The display is the same for SHOW CPU/BRIEF commands on AXP and VAX systems running SMP.

However, when executed on an AXP system, the SHOW CPU/FULL command output contains information not found in the display from a VAX SMP system. In the following VAX example, the SHOW CPU/FULL command produces a configuration summary of the VAX 6000-420 system OLEO, indicating that only CPU 02, the primary CPU, is active and in the RUN state. It also shows that there is a uniprocessing driver loaded in the system, thus preventing the system from being enabled as a multiprocessor.

```
$ ! On a VAX system
$ SHOW CPU/FULL

OLEO, A VAX 6000-420
Multiprocessing is DISABLED. MULTIPROCESSING Sysgen parameter = 02
Minimum multiprocessing revision levels -- CPU: 0 uCODE: 0 UWCS: 21.
PRIMARY CPU = 02
*** Loaded unmodified device drivers prevent multiprocessor operation.***
    RBDRIVER

CPU 02 is in RUN state
Current Process: Koko                PID = 2A6001E3
Revision levels: CPU: 0 uCODE: 0 UWCS: 0.
Capabilities of this CPU:
    PRIMARY VECTOR RUN
Processes which can only execute on this CPU:
    CONFIGURE          PID = 2A40010B Reason = PRIMARY Capability
                       Reason = RUN Capability

CPU 07 is in INIT state
Current Process: *** None ***
Revision levels: CPU: 0 uCODE: 0 UWCS: 0.
Capabilities of this CPU:
    *** None ***
Processes which can only execute on this CPU:
    *** None ***
```

In comparison, the following SHOW CPU/FULL display is from a five-CPU AXP system:

```
$ ! On an AXP system
$ SHOW CPU/FULL

LSR4, a DEC 7000 Model 650
Multiprocessing is ENABLED. Streamlined synchronization image loaded.
Minimum multiprocessing revision levels: CPU = 1

System Page Size = 8192
System Revision Code = A01
System Serial Number = 123456
Default CPU Capabilities:
    QUORUM RUN
Default Process Capabilities:
    QUORUM RUN

PRIMARY CPU = 00
```

System Setup Tasks

2.2 Setup Tasks That Are Different

```
CPU 00 is in RUN state
Current Process: *** None ***
Serial Number: 1234567
Revision:
VAX floating point operations supported.
IEEE floating point operations and data types supported.
PALCODE: Revision Code = 5.44
  PALcode Compatibility = 1
  Maximum Shared Processors = 8
  Memory Space: Physical address = 00000000 00000000
  Length = 16
  Scratch Space: Physical address = 00000000 00020000
  Length = 16
Capabilities of this CPU:
  PRIMARY QUORUM RUN
Processes which can only execute on this CPU:
  CONFIGURE      PID = 00000404  Reason: PRIMARY Capability

CPU 01 is in INIT state
Current Process: *** None ***
Serial Number: SG235LUF74
Revision: L06
VAX floating point operations supported.
IEEE floating point operations and data types supported.
PALCODE: Revision Code = 5.44
  PALcode Compatibility = 1
  Maximum Shared Processors = 8
  Memory Space: Physical address = 00000000 00000000
  Length = 16
  Scratch Space: Physical address = 00000000 00020000
  Length = 16
Capabilities of this CPU:
  *** None ***
Processes which can only execute on this CPU:
  *** None ***

CPU 02 is in RUN state
Current Process: VMSADU      PID = 00000411
Serial Number: GA24847575
Revision: 06
VAX floating point operations supported.
IEEE floating point operations and data types supported.
PALCODE: Revision Code = 5.44
  PALcode Compatibility = 1
  Maximum Shared Processors = 8
  Memory Space: Physical address = 00000000 00000000
  Length = 16
  Scratch Space: Physical address = 00000000 00020000
  Length = 16
Capabilities of this CPU:
  QUORUM RUN
Processes which can only execute on this CPU:
  *** None ***
```

System Setup Tasks

2.2 Setup Tasks That Are Different

```
CPU 04 is in RUN state
Current Process: *** None ***
Serial Number: GA24847577
Revision: 06
VAX floating point operations supported.
IEEE floating point operations and data types supported.
PALCODE: Revision Code = 5.44
  PALcode Compatibility = 1
  Maximum Shared Processors = 8
  Memory Space: Physical address = 00000000 00000000
    Length = 16
  Scratch Space: Physical address = 00000000 00020000
    Length = 16
Capabilities of this CPU:
  QUORUM RUN
Processes which can only execute on this CPU:
  *** None ***

CPU 05 is in RUN state
Current Process: *** None ***
%SMP-F-CPUTMO, CPU #01 has failed to leave the INITIALIZATION state
Serial Number: SG237LWL46
Revision:
VAX floating point operations supported.
IEEE floating point operations and data types supported.
PALCODE: Revision Code = 5.44
  PALcode Compatibility = 1
  Maximum Shared Processors = 8
  Memory Space: Physical address = 00000000 00000000
    Length = 16
  Scratch Space: Physical address = 00000000 00020000
    Length = 16
Capabilities of this CPU:
  QUORUM RUN
Processes which can only execute on this CPU:
  *** None ***

$
```

The console PALcode revision level numbers on AXP systems might be different from the numbers shown in the previous example.

2.2.12 Startup Command Procedures

As a result of the SYSMAN IO commands described in Section 2.2.10 and Appendix A, you might need to modify some of your existing `SYSS$STARTUP:*.COM` procedures *if you copy them to an OpenVMS AXP system disk*. Note, however, that the command procedures provided by OpenVMS AXP have been modified to invoke SYSMAN, instead of SYSGEN, for I/O subsystem configuration commands.

Search for AUTOCONFIGURE and update the associated command interface. For example:

```
$ SEARCH SYSS$STARTUP:*.COM AUTOCONFIGURE
```

Change SYSGEN AUTOCONFIGURE [ALL] to SYSMAN IO AUTOCONFIGURE.

2.2.13 Devices on OpenVMS AXP

Refer to the OpenVMS AXP *Software Product Description (SPD 41.87.xx)* for the most up-to-date information about the hardware devices supported with the available AXP computers.

2.2.14 Local DSA Device Naming

On OpenVMS AXP, all local DIGITAL Storage Architecture (DSA) devices use a controller letter of A, regardless of the physical controller on which the device resides. All local DSA disk devices are named DUA*n* or DJA*n*, where *n* is the unique disk unit number. All local DSA tape devices are named MUA*n*, where *n* is the unique tape unit number.

The OpenVMS AXP local device-naming scheme represents a change from OpenVMS VAX, where local DSA devices inherit the controller letter from the physical controller on which the device resides.

Table 2–7 compares the new OpenVMS AXP local DSA device-naming scheme with the local naming schemes on OpenVMS VAX and the DEC 7000 Model 600 AXP console. Note that the DEC 7000 Model 600 AXP console uses the OpenVMS VAX local DSA device-naming scheme when referring to local DSA devices. As a result, you must specify the OpenVMS VAX local DSA device names when you use the DEC 7000 Model 600 AXP console commands BOOT and SHOW DEVICE.

Table 2–7 Comparison of Device Naming on OpenVMS

Controller Where Disk Resides	OpenVMS VAX and DEC 7000 Model 600 AXP Console Local Device Naming	OpenVMS AXP Local Device Naming
PUA0	DUA0	DUA0
PUB0	DUB14	DUA14
PUC0	DUC115	DUA115

As shown in Table 2–7, OpenVMS VAX names disk unit 14 on controller PUB0 as DUB14, while OpenVMS AXP names this unit DUA14. On OpenVMS AXP, use of a single controller letter requires that the unit number for each local DSA device be unique.

Controller letters are used in device naming for hardware that artificially restricts unit number ranges. For example, Small Computer Systems Interface (SCSI) controllers currently can have disk unit numbers only from 0 through 7, which almost precludes sufficient uniqueness for any large system requiring many disks. By contrast, current DSA disks have a unit number range of 0 through 4000. In addition, the allocation class can be used to differentiate device names further. As a result, the OpenVMS AXP operating system does not add uniqueness to the device name via the controller letter.

The following benefits result from the change in local DSA device naming:

- Device naming is more uniform. Local DSA device naming is now identical to the scheme used for local DSSI devices and remote DSA devices.
- System management is simplified. Because all DSA devices now have unique unit numbers, an operator can unambiguously locate a device from among a system's disks using only the device's unit number. The operator need not be concerned whether a device with unit number 0 is DUA0 or DUB0.
- Dual pathing of a device between two OpenVMS AXP systems with local controllers is easier. Dual pathing is possible only if the device is named identically throughout the VMScluster.

System Setup Tasks

2.2 Setup Tasks That Are Different

On OpenVMS VAX, the device name inherits the controller letter from the controller on which the device resides. You must take great care to place the device on identically named controllers in each OpenVMS VAX system so that the resulting device names are identical.

With the OpenVMS AXP local DSA-naming scheme, device names are not sensitive to the controller on which the device resides, and the names always use a controller letter of A. Dual pathing can be configured without regard to the local controller on which the dual-pathed device resides.

The change in local DSA device naming in OpenVMS AXP may require that you make some changes. If local DSA devices are not already unique by unit number, you might need to reconfigure DSA devices when moving from OpenVMS VAX to OpenVMS AXP. Local DSA physical device names that are hardcoded in command files or applications may also be affected by this change.

2.2.15 File Name Format of Drivers Supplied by Digital on AXP

All drivers supplied by Digital on OpenVMS AXP use the following format:

facility-name\$xxDRIVER.EXE

The drivers included on the OpenVMS AXP kit use SYS for *facility-name*.

On OpenVMS VAX, no facility prefix is present or permitted for drivers. They are simply named xxDRIVER.EXE.

On OpenVMS AXP nodes, drivers not supplied by Digital still can be called xxDRIVER.EXE.

2.2.16 OpenVMS Installation Media

OpenVMS AXP operating system binaries and documentation are distributed on a compact disc. Other media for installations are not available. See Section B.2 for information about providing users access to the online documentation.

2.2.17 VMSINSTAL Utility

The VMSINSTAL utility on OpenVMS AXP systems includes features that are not available with the VMSINSTAL on VAX systems. This section summarizes those VMSINSTAL features.

Note

The OpenVMS VAX Version 6.1 and OpenVMS AXP Version 6.1 installations have been updated to use the POLYCENTER Software Installation utility. This product performs rapid installations and deinstallations, and lets you find information about installed products. These features are more sophisticated than the VMSINSTAL features described in this section. The key points are:

- The POLYCENTER Software Installation utility is the official replacement technology for VMSINSTAL.
- The POLYCENTER Software Installation utility is the preferred technology for packaging and shipping all layered product kits.
- VMSINSTAL will continue to ship on AXP and VAX systems for compatibility reasons.

- VMSINSTAL is no longer under active development.

See Section 2.2.2 for a summary of the POLYCENTER Software Installation utility features.

The VMSINSTAL utility on OpenVMS AXP systems contains callbacks that are not available with the VAX version of VMSINSTAL. Software developers at your site who are creating OpenVMS based software kits can read the *OpenVMS Developer's Guide to VMSINSTAL* for details.

The following VMSINSTAL utility features are of interest to system managers:

- History file of VMSINSTAL executions
- Product installation log file
- Procedure for listing installed products

2.2.17.1 History File of VMSINSTAL Executions

When VMSINSTAL terminates, a history file records the name of the product being installed and the status of the attempted installation. The history file is named SYS\$UPDATE:VMSINSTAL.HISTORY.

2.2.17.2 Product Installation Log File

If a product installation is successful using VMSINSTAL, a log file is created. This file contains information indicating:

- The product that was installed
- Who installed the product
- What files were added, deleted, modified, and so on

This file is created as SYS\$UPDATE:*facvvu*.VMI_DATA.

2.2.17.3 Procedure for Listing Installed Products

The SYS\$UPDATE:INSTALLED_PRDS.COM procedure lets the user check what products have been installed. The procedure has an optional parameter for indicating a restricted search of installed products. When executed, this procedure lists the product's name and version, when it was installed, and who installed it.

The command format is as follows:

```
@SYS$UPDATE:INSTALLED_PRDS [product-mnemonic]
```

The *product-mnemonic* value is optional. To use it, specify the save-set name of the product. If you specify *product-mnemonic*, only log files belonging to the specified product will have installation data displayed. The product mnemonic can be passed to the procedure by using any of the following search criteria:

- Product name and version (save-set name)
- Product name only
- Wildcards

System Setup Tasks

2.2 Setup Tasks That Are Different

The following command examples illustrate the installed products procedure using the search criteria:

```
$ @SYS$UPDATE:INSTALLED_PRDS
$ @SYS$UPDATE:INSTALLED_PRDS DTR010
$ @SYS$UPDATE:INSTALLED_PRDS DTR
$ @SYS$UPDATE:INSTALLED_PRDS DTR*
```

2.2.18 Running AUTOGEN

AUTOGEN is included with OpenVMS AXP. Use it to adjust the values of system parameters after installing OpenVMS AXP and after installing layered products.

The VAXVMSSYS.PAR system parameter file on OpenVMS VAX systems is called ALPHAVMSSYS.PAR on OpenVMS AXP. Like VAXVMSSYS.PAR, the ALPHAVMSSYS.PAR file resides in the SYSSSYSTEM directory.

Some system parameters are in units of pagelets, whereas others are in units of pages. AUTOGEN determines the hardware page size and records it in the PARAMS.DAT file. When reviewing AUTOGEN recommended values or when setting system parameters in MODPARAMS.DAT, note carefully which units are required for each parameter.

See Section 5.1 for information about system parameters and their units and about the tuning considerations.

2.2.19 Improving the Performance of Main Images and Shareable Images

On OpenVMS AXP, you can improve the performance of main images and shareable images that have been linked with /SHARE and the LINK qualifier /SECTION_BINDING=(CODE,DATA) by installing them as resident with the Install utility (INSTALL). The code and read-only data sections of an installed resident image can reside in granularity hint regions (GHRs) in memory. The AXP hardware can consider a set of pages as a single GHR. This GHR can be mapped by a single page table entry (PTE) in the translation buffer (TB). The result is an improvement in TB hit rates, resulting in higher performance.

Also, the OpenVMS operating system executive images are, by default, loaded into GHRs. The result is an improvement in overall OpenVMS system performance.

These options are not available on OpenVMS VAX systems.

The GHR feature lets OpenVMS split the contents of images and sort the pieces so that they can be placed with other pieces that have the same page protection in the same area of memory. Consequently, TBs on AXP systems are used more efficiently than if the loadable executive images or a user's main image or shareable images were loaded in the traditional manner.

See Section 5.5 for details.

2.2.20 SYS.EXE Renamed to SYS\$BASE_IMAGE.EXE

On OpenVMS AXP systems, the loadable executive image (SYS.EXE on VAX) has been renamed SYSSBASE_IMAGE.EXE. The file resides on the system disk in SYSSLOADABLE_IMAGES.

2.2.21 Rounding-Up Algorithm for Input Quota Values

Be careful when you assign and read the OpenVMS AXP SYSUAF process quotas that have values in pagelets (WSDEFAULT, WSQUOTA, WSEXTENT, and PGFLQUOTA). OpenVMS AXP utilities accept and display these quota values in pagelets, and then round up (if warranted). Rounding up occurs on an AXP computer with 8KB pages when the value you specify is not a multiple of 16.

For example, assume that you assign 2100 pagelets to the WSDEFAULT value for a process. On an AXP computer with 8KB pages, 2100 pagelets equal 131.25 AXP pages. The result is that AXP rounds up to 132 AXP pages. Thus, specifying 2100 pagelets is effectively the same as specifying a value in the range of 2096 to 2112 pagelets.

The AXP page-rounding operation can create interesting scenarios for system managers.

- Scenario 1
You attempt to increase slightly or decrease slightly a process quota in pagelets; in fact, no change in the number of AXP pages allocated for the process occurs internally.
- Scenario 2
You increase or decrease a process quota in terms of pagelets to a greater extent than you realized.

Scenario 1

Assume that you choose to increase slightly the WSDEFAULT value for a process. The current value is 1985 AXP pagelets, and you increase the value by 10 pagelets to 1995 pagelets. On an AXP computer with 8KB pages, 1985 pagelets equals 124.0625 AXP pages, which is rounded up internally to 125 AXP pages. The new, higher value of 1995 pagelets equals 124.6875 AXP pages, which results in the same 125 AXP pages. The net effect is that an additional working set default size was not allocated to the process, despite the command that increased the value by 10 pagelets.

Scenario 2

Assume that the PGFLQUOTA value for a process is 50000 pagelets. On an AXP computer with 8KB pages, 50000 pagelets equals 3125 AXP pages, or 25,600,000 bytes (3125 pages * 8192 bytes per page). Suppose you enter a modest increase of 10 pagelets, specifying a new PGFLQUOTA value of 50010 pagelets. On an AXP computer with 8KB pages, the 50010 pagelets equals 3125.625 AXP pages, which is rounded up to 3126 AXP pages. The 3126 AXP pages equals 25,608,192 bytes.

While you might have expected the increase of 10 pagelets to result in an additional 5120 bytes for the process PGFLQUOTA, the actual increase was 8192 bytes. The amount of the increase when AXP page boundaries are crossed would be even greater on AXP computers with 16KB, 32KB, or 64KB pages.

2.2.22 Terminal Fallback Facility

The OpenVMS Terminal Fallback Utility (TFU) is the user interface to the OpenVMS Terminal Fallback Facility (TFF). This facility provides table-driven character conversions for terminals. TFF includes a fallback driver (SYS\$FBDRIVER.EXE), a shareable image (TFFSHR.EXE), a terminal fallback utility (TFU.EXE), and a fallback table library (TFF\$MASTER.DAT).

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

System Setup Tasks

2.2 Setup Tasks That Are Different

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

- To enable fallback or to change fallback characteristics, invoke TFU as follows:

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

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

```
$ SET TERMINAL/FALLBACK
```

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

- On OpenVMS AXP, the TFF fallback driver is SYSSFBDRIVER.EXE. On OpenVMS VAX, the TFF fallback driver is FBDRIVER.EXE.
- On OpenVMS AXP, the TFF startup file is TFF\$SYSTARTUP.COM. On OpenVMS VAX, the TFF startup file is TFF\$STARTUP.COM.
- On OpenVMS AXP, TFF can handle 16-bit character fallback. The fallback table library (TFF\$MASTER.DAT) contains two more 16-bit character tables than on OpenVMS VAX. These two tables are used mainly by the Asian region. Also, the table format was changed in order to support 16-bit character fallback.
- On OpenVMS AXP, the TFU command SHOW STATISTICS does not display the size of the fallback driver (SYSSFBDRIVER.EXE).

TFF does not support RT terminals.

Refer to the *VMS Terminal Fallback Utility Manual* for more information about TFF.

Maintenance Tasks

Most OpenVMS AXP system management maintenance tasks are identical to those on OpenVMS VAX. This chapter:

- Identifies which OpenVMS system management maintenance tasks are the same on AXP and VAX
- Explains how some OpenVMS AXP system management maintenance tasks are different from those of OpenVMS VAX

3.1 Maintenance Tasks That Are the Same

Table 3–1 list the OpenVMS system management maintenance tasks that are identical or similar on AXP and VAX.

Table 3–1 Identical or Similar OpenVMS Maintenance Tasks

Feature, Task, or Command	Comments
File system	All basic file system support is present. Note that Files–11 On-Disk Structure Level 1 (ODS-1) format disks and multivolume file sets are not supported on OpenVMS AXP.
ALLOCATE command	Identical.
MOUNT command	Identical.
Accounting utility commands	Identical.
Analyzing error logs	The ANALYZE/ERROR_LOG command is the same as on OpenVMS VAX systems, with one exception: the /SUMMARY qualifier is not supported. On OpenVMS AXP, an error results if you attempt to read an error log of a VAX computer.
ANALYZE/OBJECT and ANALYZE/IMAGE commands	Command format is identical on VAX and AXP systems. You or your system's programmers should plan ahead and manage the location of native VAX VMS .OBJ and .EXE files and the location of native OpenVMS AXP .OBJ and .EXE files.
ANALYZE/PROCESS_DUMP command	Identical.
Other ANALYZE commands, /AUDIT, /CRASH_DUMP, /DISK_STRUCTURE, /MEDIA, /RMS_FILE, /SYSTEM	Identical.

(continued on next page)

Maintenance Tasks

3.1 Maintenance Tasks That Are the Same

Table 3–1 (Cont.) Identical or Similar OpenVMS Maintenance Tasks

Feature, Task, or Command	Comments
Backing up data	The BACKUP command and its qualifiers are the same. Important: Thoroughly read the <i>OpenVMS AXP Version 6.1 Release Notes</i> for the latest information about any restrictions with backup and restore operations on AXP and VAX systems.
Batch and print queuing system	The same as with OpenVMS VAX Version 6.1. Includes the multiple queue managers feature. See Section 3.2.2 for a comparison of the batch and print queuing systems on recent releases of the operating system.
CONVERT, CONVERT /RECLAIM, and the CONVSHR shareable library	Identical.
MONITOR ALL_CLASSES command	Identical. Note that VAX VMS Version 5.5 and earlier releases included the NONPAGED POOL STATISTICS class with the MONITOR ALL_CLASSES command. However, OpenVMS VAX Version 6.0 and later releases, plus OpenVMS AXP, do not include the NONPAGED POOL STATISTICS class because the former MONITOR POOL feature is not supported. This feature is replaced by adaptive pool management and two SDA commands: SHOW POOL/RING_BUFFER and SHOW POOL/STATISTICS, which are all part of recent OpenVMS releases. See Section 5.4 for more information on adaptive pool management.
MONITOR POOL command	The MONITOR POOL command, which on VMS Version 5.5 and earlier systems initiates monitoring of the NONPAGED POOL STATISTICS class and measures space allocations in the nonpaged dynamic pool, is not provided on OpenVMS AXP or on OpenVMS VAX Version 6.0 and later releases. This is due to adaptive pool management features and two SDA commands: SHOW POOL/RING_BUFFER and SHOW POOL/STATISTICS. See Section 5.4 for more information on adaptive pool management.
MONITOR MODES	The same, with one display exception: MONITOR MODES initiates monitoring of the TIME IN PROCESSOR MODES class, which includes a data item for each mode of processor operation. In displays, Interrupt Stack is replaced by Interrupt State because AXP computers do not have an interrupt stack, and service interrupts occur on the current process's kernel stack.
MONITOR/RECORD and MONITOR/INPUT	Identical. Also, MONITOR/INPUT on an AXP node can read a MONITOR.DAT file created by MONITOR/RECORD on a VAX node, and vice versa.
MONITOR TRANSACTION	Identical.
MONITOR VECTOR	Displays zeros for any AXP processor, where vectors are not supported. On an AXP computer, MONITOR VECTOR operates the same as on a VAX computer without vector processing.

(continued on next page)

Maintenance Tasks

3.1 Maintenance Tasks That Are the Same

Table 3–1 (Cont.) Identical or Similar OpenVMS Maintenance Tasks

Feature, Task, or Command	Comments
Other MONITOR commands	The following commands are the same: MONITOR CLUSTER, MONITOR DECNET, MONITOR DISK, MONITOR DLOCK, MONITOR FCP, MONITOR FILE_SYSTEM_CACHE, MONITOR IO, MONITOR LOCK, MONITOR PAGE, MONITOR PROCESSES, MONITOR RMS, MONITOR STATES, MONITOR SYSTEM.
Defragmenting disks	The OpenVMS movefile subfunction, which lets programmers write atomic-file disk-defragmentation applications, is supported on OpenVMS VAX Version 5.5 and later releases, and on OpenVMS AXP Version 6.1 and later releases. The DEC File Optimizer for OpenVMS layered product, which defragments disks using the movefile subfunction, also is supported on OpenVMS VAX Version 6.1 and on OpenVMS AXP Version 6.1.
SUMSLP	Identical.
SYSMAN utility	Similar utility functions; however, the I/O subsystem configuration functions from the OpenVMS VAX SYSGEN utility are now in the OpenVMS AXP SYSMAN utility. See Section 2.2.10 and Appendix A for details.
System Dump Analyzer (SDA)	All .STB files that are available to the System Dump Analyzer (SDA) on OpenVMS VAX are available on OpenVMS AXP systems. (Note: the .STB files are in SYSSLOADABLE_IMAGES and not in SYSSSYSTEM.) System dump file size requirements are higher on OpenVMS AXP systems. Also, there are new Crash Log Utility Extractor (CLUE) commands on OpenVMS AXP, and CLUE is part of SDA. See Section 3.2.3 for information about SDA differences.

3.2 Maintenance Tasks That Are Different

This section describes the OpenVMS system management maintenance tasks that are different on AXP systems. The differences are:

- Starting with Version 6.1, OpenVMS AXP includes an event management utility that is not available on OpenVMS VAX systems. See Section 3.2.1.
- The batch and print queuing system for OpenVMS AXP Version 6.1 is identical to the clusterwide batch and print queuing system in OpenVMS VAX Version 6.1. See Section 3.2.2.
- Larger system dump files occur and additional values for the DUMPSTYLE system parameter are provided on OpenVMS AXP. SDA is automatically invoked (after a system crash) when you reboot the system. Also, there are new Crash Log Utility Extractor (CLUE) commands on OpenVMS AXP, and CLUE is part of SDA. See Section 3.2.3.
- No patch utility is supplied for OpenVMS AXP systems. See Section 3.2.4.

Maintenance Tasks

3.2 Maintenance Tasks That Are Different

3.2.1 DECEvent Event Management Utility

The DECEvent utility is an event management utility for the OpenVMS AXP operating system. DECEvent provides the interface between a system user and the system's event log files. The utility is invoked by entering the DCL command DIAGNOSE and lets system users create ASCII reports derived from system event entries. The format of the ASCII reports depends on the command entered on the command line interface (CLI) with a maximum character limit of 255 characters.

Event report information can be filtered by event type, date, time, and event entry number. Event report formats from full disclosure to brief informational messages can be selected. The /INCLUDE and /EXCLUDE qualifiers provide a wide range of selection criteria to narrow down the focus of event searches.

The DECEvent utility also offers an interactive command shell interface that recognizes the same commands used at the command line interface. From the interactive command shell, users can customize, change, or save system settings.

DECEvent uses the system event log file, SYS\$ERRORLOG:ERRLOG.SYS, as the default input file for event reporting, unless another file is specified.

The DECEvent event management utility provides the following report types:

- Full
- Brief
- Terse
- Summary
- FSTERR

Used with qualifiers, these report types allow a system user to view event information in several ways.

See the *OpenVMS System Manager's Manual* for further details.

3.2.2 Comparison of Batch and Print Queuing Systems

Table 3–2 compares the batch and print queuing systems for recent releases of the operating systems.

Table 3–2 Comparison of Batch and Print Queuing Systems

VMS Version 5.4	VMS Version 5.5 and OpenVMS AXP Version 1.5	OpenVMS VAX Version 6.0, OpenVMS VAX Version 6.1, and OpenVMS AXP Version 6.1
Queue manager runs on each node in a cluster; no failover.	Clusterwide operation; queue manager failover to a surviving node.	Clusterwide operation, queue manager failover to a surviving node. Option of multiple queue managers to distribute batch and print work load between VMScluster nodes (to work around CPU or memory resource shortages).

(continued on next page)

Maintenance Tasks

3.2 Maintenance Tasks That Are Different

Table 3–2 (Cont.) Comparison of Batch and Print Queuing Systems

VMS Version 5.4	VMS Version 5.5 and OpenVMS AXP Version 1.5	OpenVMS VAX Version 6.0, OpenVMS VAX Version 6.1, and OpenVMS AXP Version 6.1
Queue manager runs as part of each node's job controller process.	Queue manager and job controller functions are separate.	Queue manager and job controller functions are separate.
Shared queue database, JBCSYSQUE.DAT.	Centralized queue database: QMAN\$MASTER.DAT (master file); SYSS\$QUEUE_MANAGER.QMAN\$QUEUES (queue file), and SYSS\$QUEUE_MANAGER.QMAN\$JOURNAL (journal file).	Same centralized queue database files as in VMS Version 5.5 and OpenVMS AXP Version 1.5. For each additional queue manager, the queue database contains a queue file and journal file; format is <i>name-of-manager</i> .QMAN\$QUEUES and <i>name-of-manager</i> -.QMAN\$JOURNAL.
START/QUEUE/MANAGER command.	START/QUEUE/MANAGER command has /ON=(<i>node-list</i>) qualifier to specify order in which nodes claim the queue manager during failover.	Same as in VMS Version 5.5 and OpenVMS AXP Version 1.5 but also has /ADD and /NAME_OF_MANAGER= <i>queue-manager-name</i> to create additional queue managers and distribute work load of print and queue functions in the VMScluster.
START/QUEUE/MANAGER command has /EXTEND, /BUFFER_COUNT, /RESTART qualifiers.	Obsolete.	Obsolete.
No autostart.	Autostart feature lets you start autostart queues on a node with a single command; also lets you specify a list of nodes in the VMScluster to which a queue can fail over automatically.	Same as in VMS Version 5.5 and OpenVMS AXP Version 1.5.
INITIALIZE /QUEUE command and START /QUEUE command.	New or changed commands with autostart feature: <ul style="list-style-type: none"> • INITIALIZE /QUEUE / AUTOSTART_ ON=[(<i>node::[device] [...]</i>)] • ENABLE AUTOSTART [/QUEUES] [/ON_NODE=<i>node-name</i>] • START /QUEUE / AUTOSTART_ ON=[(<i>node::[device] [...]</i>)] • DISABLE AUTOSTART [/QUEUES] [/ON_NODE=<i>node-name</i>] 	Same as in VMS Version 5.5 and OpenVMS AXP Version 1.5; however, ENABLE AUTOSTART and DISABLE AUTOSTART also include the new /NAME_OF_MANAGER qualifier.

(continued on next page)

Maintenance Tasks

3.2 Maintenance Tasks That Are Different

Table 3–2 (Cont.) Comparison of Batch and Print Queuing Systems

VMS Version 5.4	VMS Version 5.5 and OpenVMS AXP Version 1.5	OpenVMS VAX Version 6.0, OpenVMS VAX Version 6.1, and OpenVMS AXP Version 6.1
/RETAIN qualifier can be used with INITIALIZE/QUEUE, START/QUEUE, or SET QUEUE command.	/RETAIN qualifier can also be used with PRINT, SUBMIT, or SET ENTRY command.	Same as in VMS Version 5.5 and OpenVMS AXP Version 1.5.
SHOW ENTRY command display lists job name, user name, entry number, blocks, status, and name of queue.	SHOW ENTRY command display format is slightly different, making it easier to find a job's entry number. Also, SHOW ENTRY accepts job names to narrow the display criteria.	Same as in VMS Version 5.5 and OpenVMS AXP Version 1.5.
SHOW QUEUE command display lists job name, user name, entry number, status, name of queue, and node on which job is running.	SHOW QUEUE command display format is slightly different, making it easier to find a job's entry number.	Same as in VMS Version 5.5 and OpenVMS AXP Version 1.5, but also adds /MANAGER qualifier to display information about one or more queue managers.
SUBMIT command.	SUBMIT command adds /NOTE qualifier, which lets you specify a message of up to 255 characters.	Same as in VMS Version 5.5 and OpenVMS AXP Version 1.5.
FSGETQUI lexical function.	FSGETQUI lexical function enhanced to return information about the new autostart feature.	Same as in VMS Version 5.5 and OpenVMS AXP Version 1.5, but also adds information about the new manager-specific features.
\$GETQUI and \$\$NDJBC system services.	\$GETQUI and \$\$NDJBC system services enhanced to support new batch and print queuing system.	Further enhanced due to multiple queue managers; includes new parameters.
UIC-based protection of queues; default access is System:Execute, Owner:Delete, Group:Read, and World:Write.	Same as in VMS Version 5.4.	<p>C2 security adds:</p> <ul style="list-style-type: none"> • SHOW SECURITY /CLASS=QUEUE <i>queue-name</i> • SET SECURITY /CLASS=QUEUE /ACL=(ID=<i>uic</i>, ACCESS=<i>access</i>) <i>queue-name</i> • UIC-based protection of queues; default access is System:Manage, Owner:Delete, Group:Read, and World:Submit. <p>Note that the security features in OpenVMS VAX Version 6.0 and later OpenVMS VAX releases are certified as C2 compliant by the U.S. government. Although OpenVMS AXP Version 6.1 includes the same C2 security features that are in OpenVMS VAX Version 6.1, the OpenVMS AXP C2 features have not been evaluated by the U.S. government.</p>

Multiple queue managers are useful in VMScluster environments with CPU or memory shortages. It allows you to distribute the batch and print work load across different nodes in the cluster. For example, you might create separate

Maintenance Tasks

3.2 Maintenance Tasks That Are Different

queue managers for batch queues and print queues. You could then run the batch queue manager on one node and the print queue manager on a different node.

This feature is available with the following restrictions:

- The multiple queue manager feature cannot be used in a VMSccluster until one of the following is true:
 - All the nodes are running OpenVMS Version 6.1.
 - The nodes are running OpenVMS AXP Version 6.1 and OpenVMS VAX Version 6.0.
- Once you begin using multiple queue managers, you cannot bring a node running a version earlier than OpenVMS AXP Version 6.1 or OpenVMS VAX Version 6.0 into the VMSccluster environment. Doing so might result in unexpected results and failures.
- Queues running on one queue manager cannot reference queues running on a different queue manager. For example, a generic queue running on queue manager A cannot direct jobs to an execution queue running on queue manager B. In addition, you cannot move a job from a queue on queue manager A to a queue on queue manager B.
- No more than five queue managers are allowed in a VMSccluster environment.

For further details about the new batch and print queuing systems, refer to:

- *OpenVMS System Manager's Manual*
- *OpenVMS DCL Dictionary*

3.2.3 System Dump Analyzer

Differences in the System Dump Analyzer (SDA) occur with the size of the system dump file. See Section 3.2.3.1 for more information; see Section 3.2.3.2 for a related discussion about conserving dump file space. SDA is automatically invoked (after a system crash) when you reboot the system, as discussed in Section 3.2.3.3. Also, there are new Crash Log Utility Extractor (CLUE) commands on AXP systems, and CLUE is run in SDA; see Section 3.2.3.4.

3.2.3.1 Size of the System Dump File

The location and the file name of the system dump file is the same. However, VAX and AXP system dump file size requirements differ. The following calculations apply to physical memory dump files.

On VAX systems, use the following formula to calculate the correct size for SYS\$SYSTEM:SYSDUMP.DMP:

```
size-in-blocks(SYS$SYSTEM:SYSDUMP.DMP)
= size-in-pages(physical-memory)
+ (number-of-error-log-buffers * number-of-pages-per-buffer)
+ 1 (for dump header)
```

On AXP systems, use the following formula to calculate the correct size:

```
size-in-blocks(SYS$SYSTEM:SYSDUMP.DMP)
= (size-in-pages(physical-memory) * number-of-blocks-per-page)
+ (number-of-error-log-buffers * number-of-blocks-per-buffer)
+ 2 (for dump header)
```

Maintenance Tasks

3.2 Maintenance Tasks That Are Different

3.2.3.2 Conserving Dump File Storage Space

Dump file storage space might be at a premium on OpenVMS AXP computers. For system configurations with large amounts of memory, the system dump files that are automatically created can be extremely large. For example, on a 256MB system, AUTOGEN creates a dump file in excess of 500,000 blocks.

One way to conserve dump file storage space is to provide selective dumps rather than full dumps. This is *vital* on very large memory systems.

Use the DUMPSTYLE system parameter to set the desired method for capturing system dumps on BUGCHECK. On OpenVMS VAX systems, the parameter can be set to one of two values. DUMPSTYLE can be set to one of four values on OpenVMS AXP Version 6.1. When a system fails on a VAX or AXP computer, a lot of data is printed to the operator's console (if one exists); when this step completes, only then are the memory contents written fully or selectively to the dump file. Some VAX and AXP computers might not have consoles, in which case this console data never appears.

VAX systems always have full console output. On AXP systems, the information is more complex and full console output is much longer (although it contains the same basic information). The DUMPSTYLE system parameter for OpenVMS AXP Version 6.1 includes options for shorter versions of the console output. Digital picked the values 0 and 1 for the shorter console output on OpenVMS AXP systems so that you do not have to change your DUMPSTYLE system parameter to get the default, shorter output.

Table 3–3 compares the values for the DUMPSTYLE parameter.

Table 3–3 Comparison of DUMPSTYLE System Parameter Values

Value	Meaning on OpenVMS VAX	Meaning on OpenVMS AXP Version 6.1
0	Full console output; full dump	Minimal console output; full dump
1	Full console output; selective dump	Minimal console output; selective dump
2	Does not exist	Full console output; full dump
3	Does not exist	Full console output; selective dump

On AXP and VAX systems, a SHOW command in SYSGEN lists the default value for the DUMPSTYLE system parameter as 0. However, on OpenVMS AXP, the AUTOGEN calculated value (effectively a default) is 1.

You can use the following SYSGEN commands to modify the system dump file size on large memory systems:

```
$ RUN SYS$SYSTEM:SYSGEN
SYSGEN> CREATE SYS$SYSTEM:SYSDUMP.DMP/SIZE=70000
$ @SHUTDOWN
```

After the system reboots (and *only* after a reboot), you can purge SYSDUMP.DMP.

The dump file size of 70,000 blocks is sufficient to cover about 32MB of memory. This dump file size almost always encompasses the information needed to analyze a system failure.

Maintenance Tasks

3.2 Maintenance Tasks That Are Different

3.2.3.3 SDA Automatically Invoked at System Startup

Starting with OpenVMS AXP Version 6.1, SDA is automatically invoked (after a system crash) when you reboot the system. To facilitate crash dump analysis, the SDA Crash Log Utility Extractor (CLUE) automatically captures and archives selective dump file information in a CLUE list file.

A startup command procedure initiates commands that:

- Invoke SDA
- Create a CLUE\$node_ddmmy_hhmm.LIS file
- Issue a CLUE HISTORY command to populate this .LIS file

If these files accumulate more than 5,000 blocks of storage, the oldest file will be deleted. The contents of each file is where most analysis of a system failure begins. To inhibit the running of CLUE at system startup, either rename the SYS\$STARTUP:CLUE\$SDA_STARTUP.COM file, or in the SYLOGICALS.COM file, define CLUE\$INHIBIT as /SYS TRUE.

3.2.3.4 Using SDA CLUE Commands to Obtain and Analyze Crash Dump Information

On AXP systems, SDA Crash Log Utility Extractor (CLUE) extension commands can summarize information provided by certain standard SDA commands and also provide additional detail for some SDA commands. These SDA CLUE commands can interpret the contents of the dump to perform additional analysis.

The CLUE commands are:

```
CLUE CLEANUP
CLUE CONFIG
CLUE CRASH
CLUE DEBUG
CLUE ERRLOG
CLUE HISTORY
CLUE MCHK
CLUE MEMORY
CLUE PROCESS
CLUE STACK
CLUE VCC
CLUE XQP
```

All CLUE commands can be used when analyzing crash dumps; the only CLUE commands that are not allowed when analyzing a running system are CLUE CRASH, CLUE ERRLOG, CLUE HISTORY, and CLUE STACK.

Understanding SDA CLUE

When rebooting after a system failure, SDA is automatically invoked, and SDA CLUE commands analyze and save summary information from the crash dump file in CLUE history and listing files.

The CLUE HISTORY command updates the history file pointed to by the logical name CLUESHISTORY with a one-line entry and the major crash dump summary information.

The CLUE listing file contains more detailed information about the system crash and is created in the directory pointed to by the logical name CLUE\$COLLECT. The following information is included in the CLUE listing file:

- Crash dump summary information
- System configuration

Maintenance Tasks

3.2 Maintenance Tasks That Are Different

- Stack decoder
- Page and swap files
- Memory management statistics
- Process DCL recall buffer
- Active XQP processes
- XQP cache header

The CLUE list file can be printed immediately or saved for later examination.

3.2.4 Patch Utility Not Supported

The OpenVMS VAX Patch utility (PATCH) is not supported on OpenVMS AXP because compiler performance optimizations and the AXP architecture organize the placement of instructions and data in an image in a way that makes patching impractical.

The *OpenVMS Calling Standard* defines a component of each module known as a **linkage section**. You cannot make calling standard-conformant calls to routine entry points outside of the current module (or access another module's data) without referencing the linkage section. Thus, you cannot patch image code without also patching the appropriate linkage section. Patching a linkage section is difficult if you do not know what the compiler and linker have done to establish the linkage section as it appears to the image activator. For those reasons, a patch utility is not available on OpenVMS AXP.

Security Tasks

The security features in OpenVMS AXP Version 6.1 are nearly identical to those found in OpenVMS VAX Version 6.1. Note the following exceptions:

- The C2 features in OpenVMS VAX were formally evaluated and certified by the U.S. government. Although OpenVMS AXP Version 6.1 contains these same C2 features, OpenVMS AXP Version 6.1 has not been evaluated by the U.S. government.
- OpenVMS AXP does not audit DECnet network connections.

See the *OpenVMS Guide to System Security* for details about the OpenVMS security features that are common on AXP and VAX environments.

Performance Optimization Tasks

This chapter describes the OpenVMS system management performance optimization tasks that are different on AXP systems. The differences are in the following areas:

- Impact of the larger AXP page size on system parameter units. See Section 5.1.
- Changes to the default values for a number of system parameters. See Section 5.2.
- Use of page or pagelet values by OpenVMS utilities and DCL commands. See Section 5.3.
- Adaptive pool management feature. See Section 5.4.
- Installation of suitably linked main images and shareable images in a granularity hint region for improved performance. See Section 5.5.
- The virtual I/O cache (also part of OpenVMS VAX Version 6.0 and later releases) that reduces bottlenecks and improves performance. See Section 5.6.

See the *Guide to OpenVMS AXP Performance Management* for more information on OpenVMS AXP performance considerations.

5.1 System Parameters: Measurement Change for Larger Page Size

As discussed in Section 1.2.1 and as illustrated in Figure 1–1, a VAX page is 512 bytes, and an AXP page can be 8KB, 16KB, 32KB, or 64KB. The initial set of AXP computers use a page size of 8KB (8192 bytes).

The larger page size for an AXP system required a fresh look at some of the system parameters that are measured in units of VAX pages on OpenVMS VAX. The same 512-byte quantity called a page on VAX is called a pagelet on OpenVMS AXP.

The OpenVMS VAX term *page* is ambiguous in the following ways for certain system parameters in the AXP context:

- On OpenVMS VAX, “page” sometimes is used instead of disk block.
- On OpenVMS VAX, “page” sometimes is used to express a total byte size.
- On OpenVMS VAX, “page” sometimes represents a discrete memory page, regardless of the number of bytes within the page.

Certain constraints affect how some system parameters are evaluated by the operating system. For instance, the working set control parameters can be expressed in the \$CREPRC system service. As a result, a strict interpretation of pagelet must be maintained for OpenVMS AXP users.

Performance Optimization Tasks

5.1 System Parameters: Measurement Change for Larger Page Size

The system parameters and units affected by this ambiguity fall into the following categories on OpenVMS AXP:

- Units that have changed in name only. See Section 5.1.1.
- Units that are CPU specific. See Section 5.1.2.
- Parameters that have dual values (both external and internal values). See Section 5.1.3.

5.1.1 System Parameter Units That Changed in Name Only

Table 5–1 shows the system parameters whose units have changed in name only, from “page” on OpenVMS VAX to the new, more appropriate name on OpenVMS AXP.

Table 5–1 System Parameter Units That Changed in Name Only

Parameter	Unit
ACP_DINDXCACHE	Blocks
ACP_DIRCACHE	Blocks
ACP_HDRCACHE	Blocks
ACP_MAPCACHE	Blocks
ACP_WORKSET	Pagelets
CLISYMTBL	Pagelets
CTLIMGLIM	Pagelets
CTLPAGES	Pagelets
ERLBUFFERPAGES	Pagelets
IMGIOCNT	Pagelets
PIOPAGES	Pagelets
MINWSCNT	Pure number
TBSKIPWSL	Pure number

5.1.2 CPU-Specific System Parameter Units

Table 5–2 shows the units that remain as CPU-specific pages (8KB on the initial set of AXP computers).

5.1 System Parameters: Measurement Change for Larger Page Size

Table 5–2 CPU-Specific System Parameter Units

Parameter	Unit
BORROWLIM	Pages
FREEGOAL	Pages (also made DYNAMIC)
FREELIM	Pages
GROWLIM	Pages
MPW_HILIMIT	Pages
MPW_LOLIMIT	Pages
MPW_LOWAITLIMIT	Pages
MPW_THRESH	Pages
MPW_WAITLIMIT	Pages
MPW_WRTCLUSTER	Pages
GBLPAGFIL	Pages
RSRVPAGCNT	Pages

5.1.3 System Parameters with Dual Values

Table 5–3 shows the parameter units that have dual values. The parameter units in this category have both an external value and an internal value on OpenVMS AXP.

Table 5–3 System Parameters with Dual Values

Parameter	External Unit	Internal Unit	Function
PAGTBLPFC	Pagelets	Pages	Default page table page fault cluster size. Specifies the maximum number of page tables to attempt to read to satisfy a fault for a nonresident page table.
PFCDEFAULT	Pagelets	Pages	Default page fault cluster size. During execution of programs on AXP systems, controls the number of image pagelets (pages, internally) read from disk per I/O operation when a page fault occurs. The value should not be greater than one-fourth the default size of the average working set to prevent a single page fault from displacing a major portion of a working set. Too large a value for PFCDEFAULT can hurt system performance. PFCDEFAULT can be overridden on an image-by-image basis with the CLUSTER option of the OpenVMS linker.
SYSPFC	Pagelets	Pages	Page fault cluster for system paging. The number of pagelets (pages, internally) read from disk on each system paging operation.
GBLPAGES	Pagelets	Pages	Global page table entry (PTE) count. Establishes the size in pagelets (pages, internally) of the global page table and the limit for the total number of global pages that can be created.

(continued on next page)

Performance Optimization Tasks

5.1 System Parameters: Measurement Change for Larger Page Size

Table 5–3 (Cont.) System Parameters with Dual Values

Parameter	External Unit	Internal Unit	Function
SYSMWCNT	Pagelets	Pages	System working set count. Establishes the number of pagelets (pages, internally) for the working set containing the currently resident pages of pageable system space.
WSMAX	Pagelets	Pages	Maximum size of process working set. Determines the systemwide maximum size of a process working set, regardless of process quota.
VIRTUALPAGECNT	Pagelets	Pages	Maximum virtual page count. Determines the total number of pagelets (pages, internally) that can be mapped for a process, which can be divided in any fashion between P0 and P1 space.
WSINC	Pagelets	Pages	Working set increment. Sets the size in pagelets (pages, internally) to increase the working set size to compensate for a high page fault rate.
WSDEC	Pagelets	Pages	Working set decrement. Sets the number of pagelets (pages, internally) to decrease the working set to compensate for a page fault rate below the lower threshold.
AWSMIN	Pagelets	Pages	Establishes the lowest number of pagelets (pages, internally) to which a working set limit can be decreased by automatic working set adjustment.
SWPOUTPGCNT	Pagelets	Pages	Desired process page count for an outswap swap. This parameter sets the number of pagelets (pages, internally) to attempt to reduce a working set to before starting the outswap.
PQL_DPGFLQUOTA	Pagelets	Pages	Default paging file quota.
PQL_MPGFLQUOTA	Pagelets	Pages	Minimum paging file quota.
PQL_DWSDEFAULT	Pagelets	Pages	Default working set default size.
PQL_MWSDEFAULT	Pagelets	Pages	Minimum working set default size.
PQL_DWSQUOTA	Pagelets	Pages	Default working set quota.
PQL_MWSQUOTA	Pagelets	Pages	Minimum working set quota.
PQL_DWSEXTENT	Pagelets	Pages	Default working set extent.
PQL_MWSEXTENT	Pagelets	Pages	Minimum working set extent.

The external value is expressed in pagelets, and is accepted as input in \$CREPRC or returned by the \$GETSYI system service. Both SYSGEN and conversational bootstrap display both the internal and external parameter values. For example, the following is an edited SYSGEN display on OpenVMS AXP:

Performance Optimization Tasks

5.1 System Parameters: Measurement Change for Larger Page Size

Parameter Name	Current	Default	Min.	Max.	Unit	Dynamic
PFCDEFAULT	128	128	0	2032	Pagelets	D
internal value	8	8	0	127	Pages	D
GBLPAGES	443040	30720	10240	-1	Pagelets	
internal value	27690	1920	640	-1	Pages	
SYSMWCNT	5288	2048	512	65536	Pagelets	
internal value	331	128	32	4096	Pages	
WSMAX	32768	4096	1024	1048576	Pagelets	
internal value	2048	256	64	65536	Pages	
VIRTUALPAGECNT	139072	65536	2048	1000000	Pagelets	
internal value	8692	4096	128	62500	Pages	
WSINC	1200	2400	0	-1	Pagelets	D
internal value	75	150	0	-1	Pages	D
WSDEC	250	250	0	-1	Pagelets	D
internal value	16	16	0	-1	Pages	D
AWSMIN	512	512	0	-1	Pagelets	D
internal value	32	32	0	-1	Pages	D
SWOUTPGCNT	512	512	0	-1	Pagelets	D
internal value	32	32	0	-1	Pages	D
PQL_DPGFLQUOTA	65536	65536	-1	-1	Pagelets	D
internal value	4096	4096	4096	-1	Pages	D
PQL_MPGFLQUOTA	65536	2048	-1	-1	Pagelets	D
internal value	4096	128	128	-1	Pages	D
PQL_DWSDEFAULT	2000	2000	-1	-1	Pagelets	
internal value	125	125	125	-1	Pages	
PQL_MWSDEFAULT	2000	2000	-1	-1	Pagelets	
internal value	125	125	125	-1	Pages	
PQL_DWSQUOTA	6000	4000	-1	-1	Pagelets	D
internal value	375	250	250	-1	Pages	D
PQL_MWSQUOTA	4000	4000	-1	-1	Pagelets	D
internal value	250	250	250	-1	Pages	D
PQL_DWSEXTENT	65500	12000	-1	-1	Pagelets	D
internal value	4094	750	750	-1	Pages	D
PQL_MWSEXTENT	6000	4000	-1	-1	Pagelets	D
internal value	375	250	250	-1	Pages	D

Notice how the system parameter external default values (those in units of pagelets) are always multiples of 16 on an AXP system with 8KB pages. When a user specifies a pagelet value, that value is rounded up internally (if necessary) to the next whole page count because the operating system uses them in units of whole pages only, where each 8KB AXP memory page consists of 16 pagelets.

This characteristic has an important effect on system tuning. For example, you can increase a given parameter's external value by a single pagelet but not observe any effect on the behavior of the system. Because each AXP memory page consists of 16 pagelets, the parameter must be adjusted in multiples of 16 in order to change the internal value used by the operating system.

Refer to Section 2.2.21 for a related discussion of the rounding-up process with process quotas. Also, see Figure 1-1 for an illustration of the relative sizes of a VAX page, an AXP pagelet, and an AXP 8KB page.

5.2 Comparison of System Parameter Default Values

Table 5-4 shows the OpenVMS AXP system parameters whose default values, as noted in SYSGEN, are different from the value on OpenVMS VAX.

Performance Optimization Tasks

5.2 Comparison of System Parameter Default Values

Note

Table 5–4 does not repeat the OpenVMS AXP system parameters listed in Table 5–3 when the only difference is in the name of the unit (a 512-byte VAX page to a 512-byte AXP pagelet). For example, the PFCDEFAULT default value on a VAX system is 32 pages; its default value on an AXP system is 32 pagelets, the same quantity.

Also, as you compare the columns, remember:

- Each AXP pagelet is the same quantity as each VAX page (512 bytes).
- Each CPU-specific AXP page (8192 bytes per page on AXP computers with 8KB pages) is 16 times larger than each VAX page (512 bytes per page).

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

Table 5–4 Comparison of System Parameter Default Values

Parameter	VAX Value	AXP Value
GBLPAGES	15000 pages	30720 pagelets
GBLPAGFIL	1024 pages	128 pages ¹
SYSMWCNT	508 pages	2048 pagelets
BALSETCNT	16 slots	30 slots
WSMAX	1024 pages	4086 pagelets
NPAGEDYN	620000 bytes	524288 bytes
PAGEDYN	214100 bytes	212992 bytes
VIRTUALPAGECNT	12032 pages	65536 pagelets
SPTREQ	3900 pages	Obsolete
MPW_WRTCLUSTER	120 pages	64 pages
MPW_HILIMIT	500 pages	512 pages
MPW_LOLIMIT	32 pages	16 pages
MPW_THRESH	200 pages	16 pages
MPW_WAITLIMIT	620 pages	576 pages
MPW_LOWAITLIMIT	380 pages	448 pages
AWSMIN	50 pages	512 pagelets
SWPOUTPGCNT	288 pages	512 pagelets
MAXBUF	2064 bytes	8192 bytes
CLISYMTBL	250 pages	512 pagelets
LNMSHASHTBL	128 entries	512 entries
LNMPHASHTBL	128 entries	512 entries
PQL_DBIOLM	18 I/Os	32 I/Os

¹Notice that 128 AXP pages (8192 bytes per page) are twice as large as 1024 VAX pages (512 bytes per page).

(continued on next page)

Performance Optimization Tasks

5.2 Comparison of System Parameter Default Values

Table 5–4 (Cont.) Comparison of System Parameter Default Values

Parameter	VAX Value	AXP Value
PQL_DBYTLM	8192 bytes	65536 bytes
PQL_DDIOLM	18 I/Os	32 I/Os
PQL_DFILLM	16 files	128 files
PQL_DPGFLQUOTA	8192 pages	65536 pagelets
PQL_MPGFLQUOTA	512 pages	2048 pagelets
PQL_DPRCLM	8 processes	32 processes
PQL_DTQELM	8 timers	16 timers
PQL_DWSDEFAULT	100 pages	1024 pagelets
PQL_MWSDEFAULT	60 pages	512 pagelets
PQL_DWSQUOTA	200 pages	2048 pagelets
PQL_MWSQUOTA	60 pages	1024 pagelets
PQL_DWSEXTENT	400 pages	16384 pagelets
PQL_MWSEXTENT	60 pages	2048 pagelets
PQL_DENQLM	30 locks	64 locks

Note

SYSGEN lists the DUMPSTYLE default value as 0, the same value as on a VAX system. A value of 0 specifies that the entire contents of physical memory will be written to the dump file. However, on OpenVMS AXP the AUTOGEN calculated value is 1. A value of 1 specifies that the contents of memory will be written to the dump file selectively to maximize the utility of the dump file while conserving disk space.

5.3 Use of Page or Pagelet Values in Utilities and Commands

Section 1.2.1 describes the relative sizes of a VAX page (512 bytes), an AXP pagelet (also 512 bytes), and a CPU-specific AXP page (8192 bytes on an AXP computer with 8KB pages). Section 5.1 and Section 5.2 explain the impact on system parameters.

In addition to process quotas and system parameters, the page and pagelet units affect other OpenVMS system management utilities and commands, as explained in the following list:

- SHOW MEMORY

The Physical Memory Usage and Granularity Hint Regions statistics are shown in CPU-specific page units. Also, the Paging File Usage statistics are shown in blocks (rather than in 512-byte pages, as on VAX). For example:

```
$ SHOW MEMORY

System Memory Resources on 16-DEC-1994 13:46:41.99

Physical Memory Usage (pages):  Total      Free      In Use    Modified
Main Memory (96.00Mb)          12288     10020     2217      51

Virtual I/O Cache (Kbytes):    Total      Free      In Use
Cache Memory                   3200         0       3200
```

Performance Optimization Tasks

5.3 Use of Page or Pagelet Values in Utilities and Commands

Granularity Hint Regions (pages):	Total	Free	In Use	Released
Execlet code region	512	0	271	241
Execlet data region	128	1	63	64
VMS exec data region	200	3	197	0
Resident image code region	512	0	258	254
Slot Usage (slots):	Total	Free	Resident	Swapped
Process Entry Slots	119	110	9	0
Balance Set Slots	117	110	7	0
Dynamic Memory Usage (bytes):	Total	Free	In Use	Largest
Nonpaged Dynamic Memory	1196032	846080	349952	674496
Paged Dynamic Memory	1294336	826368	467968	825888
Paging File Usage (blocks):		Free	Reservable	Total
DISK\$AXPVMSSYS:[SYS0.SYSEXE]SWAPFILE.SYS		15104	15104	15104
DISK\$AXPVMSSYS:[SYS0.SYSEXE]PAGEFILE.SYS		204672	184512	204672

Of the physical pages in use, 1789 pages are permanently allocated to VMS.

- **SHOW SYSTEM**

On OpenVMS VAX systems, the SHOW SYSTEM output's rightmost column, Ph.Mem, shows the physical working set in 512-byte VAX page units. On OpenVMS AXP systems, the SHOW SYSTEM/FULL command displays CPU-specific pages *and kilobytes* in the rightmost column, Pages. For example:

```
$ ! On an AXP node
$ SHOW SYSTEM/FULL/NODE=VAXCPU

OpenVMS V5.5-2 on node VAXCPU 22-AUG-1994 13:02:59.33 Uptime 12 19:46:39
  Pid  Process Name  State Pri  I/O  CPU  Page flts  Pages
2180501A DMILLER      HIB   8    310  0 00:00:03.91  1313  307
      [VMS,DMILLER]
21801548 _RTA1:      LEF   4     59  0 00:00:00.85   373  272
      [TEST_OF_LONG_USER_IDENTIFIERS_G,TEST_OF_LONG_USER_IDENTIFIER 136Kb
```

Notes on the previous example:

- One kilobyte (KB) equals 1024 bytes. Because the previous SHOW SYSTEM/FULL command displays pages from a VAX node's processes (where each of the 307 pages equals 512 bytes, or half of 1024) with /NODE, the utility halves the 307 pages for a resulting value of 153.5KB. This value is truncated to 153KB.
- The second line for each process is displayed only when the /FULL qualifier is specified.
- Long user identifiers are truncated to 61 characters, from a maximum of 65.

The Pages column also shows CPU-specific pages and kilobytes when you use SHOW SYSTEM/FULL on an AXP node and you are displaying process information from the same or another AXP node in the VMScluster. In this case, each 8192-byte page equals 8KB. If the SHOW SYSTEM/FULL command displayed information about a process with 221 AXP pages, the value beneath it would be 1768KB (221*8).

The SHOW SYSTEM command on OpenVMS AXP displays "OpenVMS" and the version number in the banner and does not display "VAX" or "AXP."

Performance Optimization Tasks

5.3 Use of Page or Pagelet Values in Utilities and Commands

- **SHOW PROCESS/CONTINUOUS**

The Working set and Virtual pages columns show data in CPU-specific page units. The following edited output shows a snapshot of the display from a SHOW PROCESS/CONTINUOUS command:

```
$ SHOW PROCESS/CONTINUOUS

                                Process SMART                                09:52:11
State                            CUR                                Working set                    108
Cur/base priority    6/4                                Virtual pages                    447
.
.
.
$65$DKB0:[SYS0.SYSCOMMON.][SYSEXE]SHOW.EXE
```

- **MONITOR PROCESSES**

The PAGES column displays data in CPU-specific page units. For example:

```
$ MONITOR PROCESSES

Process Count: 26                VMS Monitor Utility           Uptime: 16 21:26:35
                                PROCESSES
                                on node SAMPLE
                                17-NOV-1994 09:58:48

      PID  STATE PRI  NAME          PAGES      DIOCNT  FAULTS  CPU TIME
2D000101 HIB   16 SWAPPER      0/0         0        0 00:00:00.5
2D000105 HIB   10 CONFIGURE   0/22        10       36 00:28:30.1
2D000106 HIB    7 ERRFMT     0/49       7907     35 00:00:21.8
2D000107 HIB   16 CACHE_SERVER 0/31        468     22 00:00:00.2
2D00010A HIB   10 AUDIT_SERVER 11/86       130    172 00:00:02.5
.
.
.
```

- **Ctrl/T key sequence**

The MEM field displays the current physical memory for the interactive process in CPU-specific page units. For example:

```
$ Ctrl T

SAMPLE::SMART 10:01:44 (DCL) CPU=00:00:08.88 PF=5446 IO=4702 MEM=896
```

- **SHOW PROCESS/ALL**

Under the Process Quotas category, the Paging file quota value is in pagelet units.

Under the Accounting information category, Peak working set size and Peak virtual size are in pagelet units.

Under the Process Dynamic Memory Area category, “Current Size (pagelets)” is in pagelet units.

Performance Optimization Tasks

5.3 Use of Page or Pagelet Values in Utilities and Commands

For example:

```
$ SHOW PROCESS/ALL
17-NOV-1994 09:55:47.37  User: SMART          Process ID: 2D000215
                          Node: SAMPLE          Process name: "SMART"
.
.
.
Process Quotas:
Account name: DOC
CPU limit:                Infinite  Direct I/O limit:    100
Buffered I/O byte count quota: 99808  Buffered I/O limit:  100
Timer queue entry quota:   10       Open file quota:     99
Paging file quota:        98272  Subprocess quota:    10
Default page fault cluster: 64       AST quota:           98
Enqueue quota:            600     Shared file limit:   0
Max detached processes:   0       Max active jobs:     0

Accounting information:
Buffered I/O count:       4059  Peak working set size: 3952
Direct I/O count:        380     Peak virtual size:    16688
Page faults:             5017  Mounted volumes:      0
Images activated:        63
Elapsed CPU time:        0 00:00:08.19
Connect time:            5 18:35:37.35
.
.
.
Process Dynamic Memory Area
Current Size (bytes)      57344  Current Size (pagelets) 112
Free Space (bytes)       40940  Space in Use (bytes)    16404
Size of Largest Block    40812  Size of Smallest Block  8
Number of Free Blocks     6       Free Blocks LEQU 64 Bytes 5
.
.
.
```

- **SET WORKING_SET/QUOTA**

The working set quota value that you can specify on the command line is in pagelet units. For example:

```
$ SET WORKING_SET/QUOTA=6400
%SET-I-NEWLIMS, new working set:  Limit = 150  Quota = 6400  Extent = 700
```

- **SHOW WORKING_SET**

The displayed working set values for /Limit, /Quota, /Extent, Authorized Quota, and Authorized Extent are in pagelet units *and in CPU-specific page units*:

```
$ SHOW WORKING_SET
Working Set (pagelets) /Limit=2000 /Quota=4000 /Extent=6000
Adjustment enabled    Authorized Quota=4000 Authorized Extent=6000
Working Set (8Kb pages) /Limit=125 /Quota=250 /Extent=375
Authorized Quota=250 Authorized Extent=375
```

Page units are shown in addition to the pagelet units because SHOW PROCESS and MONITOR PROCESSES commands display CPU-specific pages.

5.3 Use of Page or Pagelet Values in Utilities and Commands

- The following command qualifiers accept pagelet values:
 - RUN (process) /EXTENT /PAGE_FILE /WORKING_SET /MAXIMUM_WORKING_SET
 - SET QUEUE /WSDEFAULT /WSEXTENT /WSQUOTA
 - INITIALIZE/QUEUE /WSDEFAULT /WSEXTENT /WSQUOTA
 - START/QUEUE /WSDEFAULT /WSEXTENT /WSQUOTA
 - SET ENTRY /WSDEFAULT /WSEXTENT /WSQUOTA
 - SUBMIT /WSDEFAULT /WSEXTENT /WSQUOTA

When you or users on the AXP computer assign pagelets to the appropriate DCL command qualifiers, keep in mind the previously stated caveats—as noted in Section 2.2.21 and Section 5.1—about how pagelet values that are not multiples of 16 (on an AXP computer with 8KB pages) are rounded up to whole AXP pages.

5.4 Adaptive Pool Management

Adaptive pool management offers the following advantages:

- Simplified system management
- Improved performance
- Reduced overall pool memory requirements and less frequent denial of services because of exhaustion of resources

Note

Adaptive pool management is available on systems running any version of OpenVMS AXP, or OpenVMS VAX Version 6.0 and later releases. The feature is not available on systems running VMS Version 5.5 and earlier releases.

Adaptive pool management provides dynamic lookaside lists plus reclamation policies for returning excess packets from the list to general nonpaged pool. Note that:

- Functional interfaces to existing routines remain unchanged.
- The basic nonpaged pool granularity is increased to 64 bytes. This quantity is justified by performance studies that show it to be the optimal granularity. This increase makes the effective natural alignment 64 bytes. The consumer of nonpaged pool can continue to assume 16-byte natural alignment.
- There are 80 lookaside lists spanning an allocation range of 1 to 5120 bytes in 64-byte increments. The lists are not prepopulated; that is, they start out empty. When an allocation for a given list's size fails because the list is empty, allocation from general pool occurs. When the packet is deallocated, it is added to the lookaside list for that packet's size. Thus, the lookaside lists self-populate over time to the level needed by the average work load on the system.

Performance Optimization Tasks

5.4 Adaptive Pool Management

- The lookaside lists have a higher hit rate because of the increased number of sizes to which requests must be matched. The OpenVMS AXP method incurs 5 to 10 times fewer requests to general pool than the VAX VMS Version 5.*n* method.
- Adaptive pool management eliminates the four separate virtual regions of system space for nonpaged pool (three for lookaside lists, one for general pool). Instead, there is one large virtual region. The lookaside lists are populated from within this large region. A packet might be in one of the following states:
 - Allocated
 - Free in general pool
 - Free on some lookaside lists
- Overall memory consumption is approximately 5% less than with the VAX VMS Version 5.*n* method.

“Gentle” reclamation keeps the lists from growing too big as the result of peaks in system usage. Every 30 seconds, each of the 80 lookaside lists is examined. For each one that has at least two entries, one entry is removed to a scratch list. After each scan, a maximum of 80 entries are in the scratch list, one from each list. The scratch list entries are then returned to general pool.

“Aggressive” reclamation is triggered as a final effort to avoid pool extension from the variable allocation routine EXE\$ALONPAGVAR. The lookaside list does not have to contain at least two entries to have one removed in the scan. Even if removal would leave the list empty, the entry is removed.

The adaptive pool management feature maintains usage statistics and extends detection of pool corruption. Two versions of the SYSTEM_PRIMITIVES executive image are provided that give you a boot-time choice of a minimal pool-code version or a pool-code version that features statistics and corruption detection:

- POOLCHECK zero (default value)
SYSTEM_PRIMITIVES_MIN.EXE is loaded.
- POOLCHECK nonzero
SYSTEM_PRIMITIVES.EXE, pool checking, and monitoring version are loaded.

With the pool monitoring version loaded, the pool management code maintains a ring buffer of the most recent 256 nonpaged pool allocation and deallocation requests. Two new SDA commands are added. The following command displays the history buffer:

```
SDA> SHOW POOL/RING_BUFFER
```

The following command displays the statistics for each lookaside list:

```
SDA> SHOW POOL/STATISTICS
```

With the addition of these two SDA commands, the MONITOR POOL command is not provided on any version of OpenVMS AXP, or on OpenVMS VAX Version 6.0 and later releases.

5.5 Installing Main Images and Shareable Images In GHRs

On OpenVMS AXP, you can improve the performance of main images and shareable images that have been linked with /SHARE and the LINK qualifier /SECTION_BINDING=(CODE,DATA) by installing them as resident with the Install utility (INSTALL).

These options are not available on OpenVMS VAX systems.

The code sections and read-only data sections of an installed resident image reside in sections of memory consisting of multiple pages mapped by a single page table entry. These sections are known as granularity hint regions (GHRs). The AXP hardware can consider a set of pages as a single GHR because the GHR can be mapped by a single page table entry (PTE) in the translation buffer (TB). The result is an increase in TB hit rates. Consequently, TBs on AXP systems are used more efficiently than if the loadable executive images or the shareable images were loaded in the traditional manner.

Also, the OpenVMS AXP operating system executive images are, by default, loaded into GHRs. The result is that overall OpenVMS system performance is improved. This feature is controlled by system parameters, as discussed in Section 5.5.2.

The GHR feature lets OpenVMS split the contents of images and sort the pieces so that they can be placed with other pieces that have the same page protection in the same area of memory. This method enables a single PTE to map the multiple pages.

Application programmers are the likely users of the GHR feature for shareable images. As system manager, you might be asked to coordinate or assist GHR setup efforts by entering Install utility and SYSGEN commands.

The CODE keyword in the LINK/SECTION_BINDING=*option* command indicates that the linker should not optimize calls between code image sections by using a relative branch instruction.

The DATA keyword indicates that the linker must ensure that no relative references exist between data image sections. If the image is linked with the DATA option on the /SECTION_BINDING qualifier, the image activator compresses the data image sections in order not to waste virtual address space. While it does save virtual address space, the DATA option to the /SECTION_BINDING qualifier does not improve performance.

See the *OpenVMS Linker Utility Manual* for details about the /SECTION_BINDING qualifier.

You can use the ANALYZE/IMAGE command on an OpenVMS AXP system to determine whether an image was linked with /SECTION_BINDING=(CODE,DATA). In the ANALYZE/IMAGE output, look for the EIHDSV_BIND_CODE and EIHDSV_BIND_DATA symbols; a value of 1 for each symbol indicates that /SECTION_BINDING=(CODE,DATA) was used.

Performance Optimization Tasks

5.5 Installing Main Images and Shareable Images In GHRs

5.5.1 Install Utility Support

Several OpenVMS AXP Install utility (INSTALL) commands have been enhanced to support the GHR feature. The ADD, CREATE, and REPLACE commands have a new qualifier, /RESIDENT. When this qualifier is specified, INSTALL loads all image sections that have the EXE and NOWRT attributes into one of the granularity hint regions (GHRs) in system space. If no image sections have these attributes, INSTALL issues the following warning message, where X is the image name:

```
%INSTALL-W-NORESSEC, no resident sections created for X
```

Note

Use of /RESIDENT is applicable only to loading main images or shareable images.

The display produced by the INSTALL commands LIST and LIST/FULL shows those images that are installed resident. For the LIST/FULL command, the display shows how many resident code sections were found. For example:

```
INSTALL> LIST SYS$LIBRARY:FOO.EXE
FOO;1   Open Hdr Shar      Lnkbl      Resid
INSTALL> LIST/FULL
FOO;1   Open Hdr Shar      Lnkbl      Resid
      Entry access count    = 0
      Current / Maximum shared = 1 / 0
      Global section count   = 1
      Resident section count = 0001
```

Note

The LIBOTS.EXE, LIBRTL.EXE, DPML\$SHR.EXE, DECC\$SHR.EXE, and CMA\$TIS_SHR.EXE images are installed resident on OpenVMS AXP.

5.5.2 System Parameter Support

Five system parameters are associated with the GHR feature:

- GH_RSRVPGCNT
- GH_EXEC_CODE
- GH_EXEC_DATA
- GH_RES_CODE
- GH_RES_DATA

Table 5-5 summarizes the purpose of each system parameter.

Performance Optimization Tasks

5.5 Installing Main Images and Shareable Images In GHRs

Table 5–5 System Parameters Associated with GHR Feature

System Parameter	Description
GH_RSRVPGCNT	<p>Specifies the number of unused pages within a GHR to be retained after startup. The default value for GH_RSRVPGCNT is 0. At the end of startup, the LDR\$WRAPUP.EXE image executes and releases all unused portions of the GHR except for the amount specified by GH_RSRVPGCNT, assuming that the SGN\$V_RELEASE_PFNS flag is set in the system parameter LOAD_SYS_IMAGES. This flag is set by default. Setting GH_RSRVPGCNT to a nonzero value lets images be installed resident at run time.</p> <p>If there are no GH_RSRVPGCNT pages in the GHR when LDR\$WRAPUP runs, no attempt is made to allocate more memory. Whatever free space is left in the GHR will be available for use by INSTALL.</p>
GH_EXEC_CODE	Specifies the size in pages of the executable code granularity hint region.
GH_EXEC_DATA	Specifies the size in pages of the executable data granularity hint region.
GH_RES_CODE	Specifies the size in pages of the resident image code granularity hint region.
GH_RES_DATA	Specifies the size in pages of the resident image data granularity hint region.

For a listing of all system parameters, see the *OpenVMS System Management Utilities Reference Manual*.

5.5.3 SHOW MEMORY Support

The DCL command SHOW MEMORY has been enhanced to support the granularity hint region (GHR) feature. The new /GH_REGIONS qualifier displays information about the GHRs that have been established. For each of these regions, information is displayed about the size of the region, the amount of free memory, the amount of memory in use, and the amount of memory released from the region.

In addition, the GHR information is displayed as part of the SHOW MEMORY, SHOW MEMORY/ALL, and SHOW MEMORY/FULL commands.

5.5.4 Loader Changes for Executive Images in GHRs

In traditional executive image loading, code and data are sparsely laid out in system address space. The loader allocates the virtual address space for executive images so that the image sections are loaded on the same boundaries as the linker created them.

The loader allocates a granularity hint region (GHR) for nonpaged code and another for nonpaged data. Pages within a GHR must have the same protection; hence, code and data cannot share a GHR. The end result is a single TB entry to map the executive nonpaged code and another to map the nonpaged data.

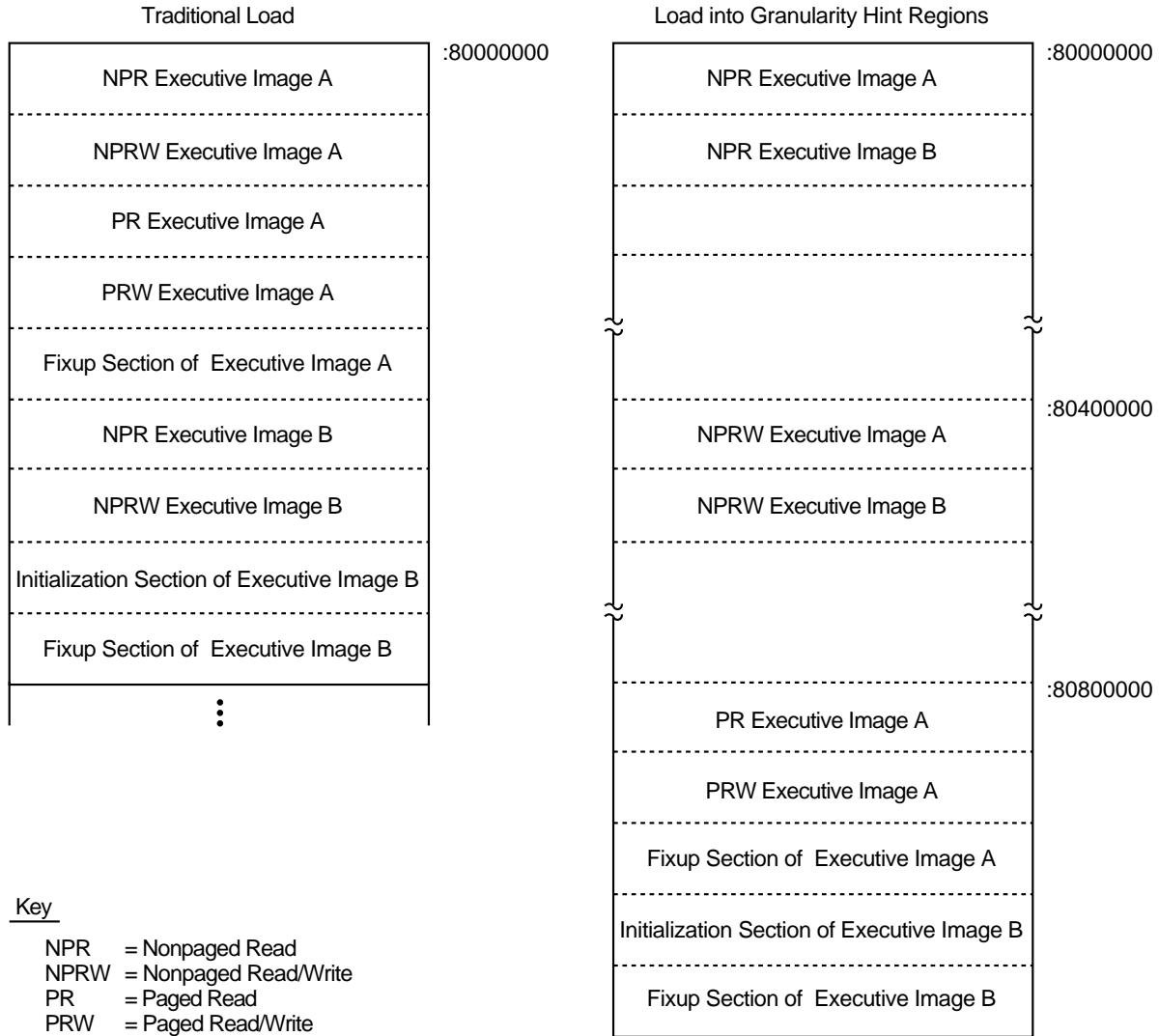
The loader then loads like nonpaged sections from each loadable executive image into the same region of virtual memory. The loader ignores the page size with which the image was linked. Paged, fixup, and initialization sections are loaded in the same manner as the traditional loader. If the S0_PAGING parameter is set to turn off paging of the executive image, all code and data, both paged and nonpaged, are treated as nonpaged and are loaded into the GHR.

Performance Optimization Tasks

5.5 Installing Main Images and Shareable Images In GHRs

Figure 5–1 illustrates a traditional load and a load into a GHR.

Figure 5–1 Traditional Loads and Loads into GHRs



ZK-5353A-GE

5.6 Virtual I/O Cache

The virtual I/O cache is a file-oriented disk cache that reduces I/O bottlenecks and improves performance. Cache operation is transparent to application software and requires little system management. This functionality provides a write-through cache that maintains the integrity of disk writes while significantly improving read performance.

By default, virtual I/O caching is enabled. To disable caching, set the system parameter `VCC_FLAGS` to 0; to enable it again, set the parameter to 1. By default, memory is allocated for caching 6400 disk blocks. This requires 3.2MB of memory. Use the `VCC_MAXSIZE` system parameter to control memory allocation.

Use the DCL commands `SHOW MEMORY/CACHE` and `SHOW MEMORY /CACHE/FULL` to observe cache statistics.

Network Management Tasks

You can use DECnet for OpenVMS AXP to establish networking connections with other nodes. DECnet for OpenVMS, previously known as DECnet-VAX on the VAX VMS platform, implements Phase IV of the Digital network architecture. The DECnet for OpenVMS AXP features are similar to those of the DECnet for OpenVMS VAX Version 6.1, with a few exceptions. This chapter:

- Identifies which OpenVMS network management tasks remain the same on AXP and VAX systems
- Explains how some network management tasks differ on OpenVMS AXP

Note

The comparisons in this chapter are specific to the Phase IV software, DECnet for OpenVMS. If your OpenVMS AXP node is using DECnet/OSI (Phase V), refer to the DECnet/OSI manuals and release notes for details.

If you install the version of DECnet/OSI that is for OpenVMS AXP Version 6.1:

- DECdns client is available.
 - X.25 support is available through DEC X.25 for OpenVMS AXP. The QIO interface drivers from the P.S.I. product are included in DEC X.25 in order to provide the same interface to customer applications.
-

6.1 Network Management Tasks That Are the Same

Table 6-1 lists the OpenVMS network management tasks that are identical or similar on AXP and VAX computers.

Table 6-1 Identical or Similar OpenVMS Network Management Tasks

Feature or Task	Comment
Product Authorization Key (PAK) names	The PAK name for the end node license (DVNETEND) is the same on AXP and VAX systems. However, the PAK name enabling cluster alias routing support on OpenVMS AXP (DVNETEXT) is different from the PAK name enabling cluster alias routing support on OpenVMS VAX (DVNETRTG). See Section 6.2.1 for related information.

(continued on next page)

Network Management Tasks

6.1 Network Management Tasks That Are the Same

Table 6–1 (Cont.) Identical or Similar OpenVMS Network Management Tasks

Feature or Task	Comment
Cluster alias	Similar; however, level 1 routing is supported only on routers for a cluster alias. See Section 6.2.1 for more information.
NETCONFIG.COM procedure	The same, with one exception. See Section 6.2.1 for more information.
NETCONFIG_UPDATE.COM procedure	Identical.
Configuring DECnet databases and starting OpenVMS AXP computer's access to the network	The process of configuring your node and starting DECnet network connections for your computer is essentially the same on OpenVMS AXP (with the limitations in Section 6.2 in mind). The functions of the SYSSMANAGER:STARTNET.COM procedure are similar.
File access listener (FAL)	FAL is fully compatible with the OpenVMS VAX version. For example, bidirectional file transfer using the COPY command, which uses the FAL object, is identical.
Maximum network size	The same Phase IV limitations for VAX nodes running DECnet for OpenVMS (1023 nodes per area and 63 areas in the entire network). Note, however, that the size of the network may be much larger if you are running DECnet/OSI for OpenVMS AXP. See your DECnet/OSI documentation for details.
Node name rules	The same rules and 6-character maximum length as with VAX nodes running DECnet for OpenVMS.
DECnet objects	Identical.
Task-to-task communication	Identical.
Network management using Network Control Program (NCP) utility and the network management listener (NML) object	In many cases, the NCP commands and parameters are identical. However, a number of NCP command parameters that are available for SET and subsequent SHOW operations have no effect on OpenVMS AXP. This characteristic is due to the lack of support for DDCCMP, full host-based routing, the Distributed Name Service (DNS), and VAX P.S.I. See Section 6.2.5 for details.
Event logging	Identical.
DECnet Test Sender/DECnet Test Receiver utility (DTS/DTR)	Identical.
Downline load and upline dump operations	Identical.
Loopback mirror testing	Identical.
Ethernet monitor (NICONFIG)	Identical.
Supported lines	Ethernet and FDDI only.
Routing	Level 1 routing is supported only on nodes acting as routers for a cluster alias. Level 2 routing and routing between multiple circuits is not supported. See Section 6.2.1 and Section 6.2.5 for details.
SET HOST capabilities	Identical.

6.2 Network Management Tasks That Are Different

This section describes the OpenVMS network management tasks that are different on AXP systems. The differences are:

- Level 2 routing (between DECnet areas) is not supported. Level 1 routing is supported only on nodes acting as routers for a cluster alias. Routing between multiple circuits is not supported. Section 6.2.1.
- Some line types are not supported. See Section 6.2.2.
- The Distributed Name Service (DNS) node name interface that is used on OpenVMS VAX is not supported on OpenVMS AXP. See Section 6.2.3.
- VAX P.S.I. is not supported. See Section 6.2.4.
- A number of NCP command parameters are affected by unsupported features. See Section 6.2.5 for details.

6.2.1 Level 1 Routing Supported for Cluster Alias Only

Level 1 DECnet routing is available, but is supported only on DECnet for OpenVMS AXP nodes acting as routers for a cluster alias. Routing between multiple circuits is not supported. Level 2 routing is not supported on DECnet for OpenVMS AXP nodes.

Note that the Product Authorization Key (PAK) name enabling DECnet for OpenVMS AXP cluster alias routing support (DVNETEXT) is different from the PAK name enabling cluster alias routing support on OpenVMS VAX (DVNETRTG). The functions supported with the DVNETEXT license differ from the DVNETRTG license. DVNETEXT is supported only to enable level 1 routing on AXP nodes acting as routers for a cluster alias.

Enabling a cluster alias requires differing steps, depending on whether you want the alias enabled for incoming, outgoing, or both types of connections. The different cases are documented in the *DECnet for OpenVMS Networking Manual* and *DECnet for OpenVMS Network Management Utilities*.

With the cluster alias feature, the difference between AXP and VAX systems is that you will have to manually enable level 1 routing on one of the AXP nodes¹ because the NETCONFIG.COM command procedure does not ask the routing question. (On VAX systems, NETCONFIG.COM prompts the user with the query, “Do you want to operate as a router?”)

Enter DEFINE commands in NCP that are similar to the following example. (If DECnet is already running, shut down DECnet, enter the DEFINE commands, then restart DECnet.) The following example enables level 1 routing on one or more nodes, identifies the node name or node address of the alias node, and allows this node to accept (ENABLED) or not accept (DISABLED) incoming connections addressed to the cluster alias:

```
$ RUN SYS$SYSTEM:NCP
NCP>DEFINE EXECUTOR TYPE ROUTING IV ! Only necessary on cluster alias routers
NCP>DEFINE NODE alias-node-address NAME alias-node-name
NCP>DEFINE EXECUTOR ALIAS NODE alias-node-name
NCP>DEFINE EXECUTOR ALIAS INCOMING {ENABLED DISABLED}
```

¹ In a VMScluster, you do not have to enable level 1 routing on an AXP node if one of the VAX VMS Version 5.5–2 nodes or OpenVMS VAX Version 6.*n* nodes is a routing node.

Network Management Tasks

6.2 Network Management Tasks That Are Different

Specifying ENABLED or DISABLED with the NCP command DEFINE EXECUTOR ALIAS INCOMING is the network manager's choice, depending on whether you want this node to accept incoming connections directed to the cluster alias. If set to DISABLED, another node in the VMScluster having this parameter set to ENABLED will handle incoming alias connections.

Note that the CLUSTER_CONFIG.COM procedure uses NCP commands to configure the cluster alias when the following two conditions exist:

- You add a new VMScluster member node.
- The system from which you are running CLUSTER_CONFIG.COM has a cluster alias defined.

See the *DECnet for OpenVMS Networking Manual* and *DECnet for OpenVMS Network Management Utilities* for details.

See Section 6.2.5 for information about NCP command parameters that are available but have no effect on OpenVMS AXP systems because level 2 routing is not supported and level 1 routing is reserved for the cluster alias.

6.2.2 CI and DDCMP Lines Not Supported

OpenVMS AXP nodes can connect to a local area network (LAN) using Ethernet lines or FDDI lines only.

DECnet communication over CI lines is not supported. There also is no support for DDCMP lines.

Because DDCMP lines are not supported, the DCL command SET TERMINAL /PROTOCOL=DDCMP/SWITCH=DECNET also is not supported on OpenVMS AXP systems.

See Section 6.2.5 for information about NCP command parameters that are available but have no effect on OpenVMS AXP systems because DDCMP is not supported.

6.2.3 DNS Node Name Interface Not Supported

With DECnet for OpenVMS AXP (Phase IV), the Distributed Name Service (DNS) node name interface that is used on OpenVMS VAX is not supported. Consequently, DNS object names cannot be specified by users and applications on OpenVMS AXP nodes running Phase IV. See Section 6.2.5 for information about NCP command parameters that are available but have no effect on OpenVMS AXP systems because a DNS is not supported.

Note that the DECdns client is supported with DECnet/OSI. See the DECnet/OSI documentation for details.

6.2.4 VAX P.S.I. Not Supported

VAX P.S.I., a software product that enables connections to X.25 networks, is not supported. See Section 6.2.5 for information about NCP command parameters that are available but have no effect on OpenVMS AXP systems because VAX P.S.I. is not supported.

Although VAX P.S.I. is not supported, a DECnet for OpenVMS AXP node can communicate with DECnet nodes that are connected to X.25 networks reachable via a DECnet/X.25 router.

Network Management Tasks

6.2 Network Management Tasks That Are Different

6.2.5 NCP Command Parameters Affected by Unsupported Features

On nodes running DECnet for OpenVMS AXP, you can set unsupported NCP command parameters and then display those settings with the SHOW command. However, such parameters have no effect on OpenVMS AXP systems and are related to the following unsupported features or products:

- DDCMP.
- VAX P.S.I.; however, a DECnet for OpenVMS AXP node can communicate with DECnet nodes that are connected to X.25 networks reachable via a DECnet/X.25 router.
- Level 2 routing. (Level 1 routing is supported only on nodes acting as routers for a cluster alias; routing between multiple circuits is not supported.)
- Distributed Name Service (DNS).

Note

The characteristic on OpenVMS AXP of being able to set and show NCP command parameters that are related to unsupported features is similar to the same characteristic on OpenVMS VAX. For example, on a VAX system you could set and show NCP command parameters related to X.25 networks even if you had not installed VAX P.S.I.

Table 6–2 lists the affected NCP command parameters related to the unsupported DDCMP circuits, VAX P.S.I. software, full host-based routing, and the DNS node name interface. Refer to the *DECnet for OpenVMS Network Management Utilities* manual for details about how these parameters are used on OpenVMS VAX computers.

Table 6–2 NCP Command Parameters Affected by Unsupported Features

NCP Command Parameter	Associated Unsupported Feature
CLEAR/PURGE CIRCUIT ACTIVE BASE	DDCMP
CLEAR/PURGE CIRCUIT ACTIVE INCREMENT	DDCMP
CLEAR/PURGE CIRCUIT BABBLE TIMER	DDCMP
CLEAR/PURGE CIRCUIT DEAD THRESHOLD	DDCMP
CLEAR/PURGE CIRCUIT DYING BASE	DDCMP
CLEAR/PURGE CIRCUIT DYING INCREMENT	DDCMP
CLEAR/PURGE CIRCUIT DYING THRESHOLD	DDCMP
CLEAR/PURGE CIRCUIT INACTIVE BASE	DDCMP
CLEAR/PURGE CIRCUIT INACTIVE INCREMENT	DDCMP
CLEAR/PURGE CIRCUIT INACTIVE THRESHOLD	DDCMP
CLEAR/PURGE CIRCUIT MAXIMUM BUFFERS	DDCMP
CLEAR/PURGE CIRCUIT MAXIMUM RECALLS	DDCMP
CLEAR/PURGE CIRCUIT MAXIMUM TRANSMITS	DDCMP

(continued on next page)

Network Management Tasks

6.2 Network Management Tasks That Are Different

Table 6–2 (Cont.) NCP Command Parameters Affected by Unsupported Features

NCP Command Parameter	Associated Unsupported Feature
CLEAR/PURGE CIRCUIT NETWORK	VAX P.S.I.
CLEAR/PURGE CIRCUIT NUMBER	VAX P.S.I.
CLEAR/PURGE CIRCUIT RECALL TIMER	VAX P.S.I.
CLEAR/PURGE CIRCUIT TRANSMIT TIMER	DDCMP
CLEAR/PURGE EXECUTOR AREA MAXIMUM COST	Host-based routing ¹
CLEAR/PURGE EXECUTOR AREA MAXIMUM HOPS	Host-based routing
CLEAR/PURGE EXECUTOR DNS INTERFACE	DNS node name interface
CLEAR/PURGE EXECUTOR DNS NAMESPACE	DNS node name interface
CLEAR/PURGE EXECUTOR IDP	DNS node name interface
CLEAR/PURGE EXECUTOR MAXIMUM AREA	Host-based routing
CLEAR/PURGE EXECUTOR MAXIMUM PATH SPLITS	Host-based routing
CLEAR/PURGE EXECUTOR PATH SPLIT POLICY	Host-based routing
CLEAR/PURGE EXECUTOR ROUTING TIMER	Host-based routing
CLEAR/PURGE EXECUTOR SUBADDRESSES	VAX P.S.I.
CLEAR/PURGE LINE DEAD TIMER	DDCMP
CLEAR/PURGE LINE DELAY TIMER	DDCMP
CLEAR/PURGE LINE HANGUP	DDCMP
CLEAR/PURGE LINE HOLDBACK TIMER	VAX P.S.I.
CLEAR/PURGE LINE LINE SPEED	DDCMP
CLEAR/PURGE LINE MAXIMUM RETRANSMITS	VAX P.S.I.
CLEAR/PURGE LINE SCHEDULING TIMER	DDCMP
CLEAR/PURGE LINE STREAM TIMER	DDCMP
CLEAR/PURGE LINE SWITCH	DDCMP
CLEAR/PURGE LINE TRANSMIT PIPELINE	DDCMP
CLEAR/PURGE MODULE X25-ACCESS (all parameters)	VAX P.S.I.
CLEAR/PURGE MODULE X25-PROTOCOL (all parameters)	VAX P.S.I.
CLEAR/PURGE MODULE X25-SERVER/X29-SERVER (all parameters)	VAX P.S.I.
CLEAR/PURGE NODE INBOUND	DDCMP
LOOP LINE (all parameters)	VAX P.S.I.
SET/DEFINE CIRCUIT ACTIVE BASE	DDCMP
SET/DEFINE CIRCUIT ACTIVE INCREMENT	DDCMP
SET/DEFINE CIRCUIT BABBLE TIMER	DDCMP
SET/DEFINE CIRCUIT CHANNEL	VAX P.S.I.
SET/DEFINE CIRCUIT DEAD THRESHOLD	DDCMP

¹Level 2 routing is not supported. Level 1 routing is supported only on nodes acting as routers for a cluster alias; routing between multiple circuits is not supported.

(continued on next page)

Network Management Tasks

6.2 Network Management Tasks That Are Different

Table 6–2 (Cont.) NCP Command Parameters Affected by Unsupported Features

NCP Command Parameter	Associated Unsupported Feature
SET/DEFINE CIRCUIT DTE	VAX P.S.I.
SET/DEFINE CIRCUIT DYING BASE	DDCMP
SET/DEFINE CIRCUIT DYING INCREMENT	DDCMP
SET/DEFINE CIRCUIT DYING THRESHOLD	DDCMP
SET/DEFINE CIRCUIT INACTIVE BASE	DDCMP
SET/DEFINE CIRCUIT INACTIVE INCREMENT	DDCMP
SET/DEFINE CIRCUIT INACTIVE THRESHOLD	DDCMP
SET/DEFINE CIRCUIT MAXIMUM BUFFERS	DDCMP
SET/DEFINE CIRCUIT MAXIMUM DATA	VAX P.S.I.
SET/DEFINE CIRCUIT MAXIMUM RECALLS	VAX P.S.I.
SET/DEFINE CIRCUIT MAXIMUM TRANSMITS	DDCMP
SET/DEFINE CIRCUIT MAXIMUM WINDOW	VAX P.S.I.
SET/DEFINE CIRCUIT NETWORK	VAX P.S.I.
SET/DEFINE CIRCUIT NUMBER	VAX P.S.I.
SET/DEFINE CIRCUIT OWNER EXECUTOR	VAX P.S.I.
SET/DEFINE CIRCUIT POLLING STATE	DDCMP
SET/DEFINE CIRCUIT RECALL TIMER	VAX P.S.I.
SET/DEFINE CIRCUIT TRANSMIT TIMER	DDCMP
SET/DEFINE CIRCUIT TRIBUTARY	DDCMP
SET/DEFINE CIRCUIT TYPE X25	VAX P.S.I.
SET/DEFINE CIRCUIT USAGE	VAX P.S.I.
SET/DEFINE CIRCUIT VERIFICATION	DDCMP
SET/DEFINE EXECUTOR AREA MAXIMUM COST	Host-based routing
SET/DEFINE EXECUTOR AREA MAXIMUM HOPS	Host-based routing
SET/DEFINE EXECUTOR DNS INTERFACE	DNS node name interface
SET/DEFINE EXECUTOR DNS NAMESPACE	DNS node name interface
SET/DEFINE EXECUTOR IDP	DNS node name interface
SET/DEFINE EXECUTOR MAXIMUM AREA	Host-based routing
SET/DEFINE EXECUTOR MAXIMUM PATH SPLITS	Host-based routing
SET/DEFINE EXECUTOR PATH SPLIT POLICY	Host-based routing
SET/DEFINE EXECUTOR ROUTING TIMER	Host-based routing
SET/DEFINE EXECUTOR SUBADDRESSES	VAX P.S.I.

(continued on next page)

Network Management Tasks

6.2 Network Management Tasks That Are Different

Table 6–2 (Cont.) NCP Command Parameters Affected by Unsupported Features

NCP Command Parameter	Associated Unsupported Feature
SET/DEFINE EXECUTOR TYPE	The AREA node-type parameter is not supported because of the lack of level 2 host-based routing. The NONROUTING IV node-type parameter is supported. The ROUTING IV node-type parameter is only supported for cluster alias routers.
SET/DEFINE LINE CLOCK	DDCMP
SET/DEFINE LINE DEAD TIMER	DDCMP
SET/DEFINE LINE DELAY TIMER	DDCMP
SET/DEFINE LINE DUPLEX	DDCMP
SET/DEFINE LINE HANGUP	DDCMP
SET/DEFINE LINE HOLDBACK TIMER	VAX P.S.I.
SET/DEFINE LINE INTERFACE	VAX P.S.I.
SET/DEFINE LINE SPEED	DDCMP
SET/DEFINE LINE MAXIMUM BLOCK	VAX P.S.I.
SET/DEFINE LINE MAXIMUM RETRANSMITS	VAX P.S.I.
SET/DEFINE LINE MAXIMUM WINDOW	VAX P.S.I.
SET/DEFINE LINE MICROCODE DUMP	VAX P.S.I.
SET/DEFINE LINE NETWORK	VAX P.S.I.
SET/DEFINE LINE RETRANSMIT TIMER	DDCMP and VAX P.S.I.
SET/DEFINE LINE SCHEDULING TIMER	DDCMP
SET/DEFINE LINE STREAM TIMER	DDCMP
SET/DEFINE LINE SWITCH	DDCMP
SET/DEFINE LINE TRANSMIT PIPELINE	DDCMP
SET/DEFINE MODULE X25-ACCESS (all parameters)	VAX P.S.I.
SET/DEFINE MODULE X25-PROTOCOL (all parameters)	VAX P.S.I.
SET/DEFINE MODULE X25-SERVER/X29-SERVER (all parameters)	VAX P.S.I.
SET/DEFINE NODE INBOUND	DDCMP
SHOW/LIST CIRCUIT display: Polling substate value	DDCMP
SHOW/LIST MODULE X25-ACCESS (all parameters)	VAX P.S.I.
SHOW/LIST MODULE X25-PROTOCOL (all parameters)	VAX P.S.I.
SHOW/LIST MODULE X25-SERVER/X29-SERVER (all parameters)	VAX P.S.I.
ZERO MODULE X25-PROTOCOL (all parameters)	VAX P.S.I.
ZERO MODULE X25-SERVER/X29-SERVER (all parameters)	VAX P.S.I.

A

I/O Subsystem Configuration Commands in SYSMAN

This appendix describes the I/O subsystem configuration support that is added to the System Management utility (SYSMAN) for OpenVMS AXP.

A.1 I/O Subsystem Configuration Support in SYSMAN

On OpenVMS AXP, SYSMAN is used to connect devices, load I/O device drivers, and display configuration information useful for debugging device drivers. I/O commands are in the System Generation utility (SYSGEN) on OpenVMS VAX.

Enter the following command to invoke SYSMAN:

```
$ SYSMAN ::= $SYS$SYSTEM:SYSMAN
SYSMAN>
```

All SYSMAN commands that control and display the I/O configuration of an OpenVMS AXP computer must be preceded by "IO". For example, to configure a system automatically, enter the following command:

```
SYSMAN> IO AUTOCONFIGURE
```

This section contains a syntax description for the SYSMAN IO commands AUTOCONFIGURE, CONNECT, LOAD, SET PREFIX, SHOW BUS, SHOW DEVICE, and SHOW PREFIX.

IO AUTOCONFIGURE

Automatically identifies and configures all hardware devices attached to a system. The IO AUTOCONFIGURE command connects devices and loads their drivers.

You must have CMKRNL and SYSLCK privileges to use the IO AUTOCONFIGURE command.

Format

```
IO AUTOCONFIGURE
```

Parameters

None.

Qualifiers

/SELECT=(device-name)

Specifies the device type to be configured automatically. Use valid device names or mnemonics that indicate the devices to be included in the configuration. Wildcards must be explicitly specified.

I/O Subsystem Configuration Commands in SYSMAN IO AUTOCONFIGURE

The /SELECT and /EXCLUDE qualifiers are not mutually exclusive as they are on OpenVMS VAX. Both qualifiers can be specified on the command line.

Table A-1 shows /SELECT qualifier examples.

Table A-1 /SELECT Qualifier Examples

Command	Configured Devices	Unconfigured Devices
/SELECT=P*	PKA,PKB,PIA	None
/SELECT=PK*	PKA,PKB	PIA
/SELECT=PKA*	PKA	PKB,PIA

/EXCLUDE=(device-name)

Specifies the device type that should not be configured automatically. Use valid device names or mnemonics that indicate the devices to be excluded from the configuration. Wildcards must be explicitly specified.

The /SELECT and /EXCLUDE qualifiers are not mutually exclusive as they are on OpenVMS VAX. Both qualifiers can be specified on the command line.

/LOG

Controls whether the IO AUTOCONFIGURE command displays information about loaded devices.

Description

The IO AUTOCONFIGURE command identifies and configures all hardware devices attached to a system. You must have CMKRNL and SYSLCK privileges to use the IO AUTOCONFIGURE command.

Examples

1. SYSMAN> IO AUTOCONFIGURE/EXCLUDE=DKA0

The command in this example autoconfigures all devices on the system except for DKA0.

IO AUTOCONFIGURE automatically configures all standard devices that are physically attached to the system, except for the network communications device.

2. SYSMAN> IO AUTOCONFIGURE/LOG

The /LOG qualifier displays information about all the devices that AUTOCONFIGURE loads.

IO CONNECT

Connects a hardware device and loads its driver, if the driver is not already loaded.

You must have CMKRNL and SYSLCK privileges to use the IO CONNECT command.

Format

IO CONNECT device-name[:]

Parameters

device-name[:]

Specifies the name of the hardware device to be connected. It should be specified in the format *device-type controller unit-number*. For example, in the designation LPA0, LP is a line printer on controller A at unit number 0. If the /NOADAPTER qualifier is specified, the device is the software device to be loaded.

Qualifiers

/ADAPTER=tr-number

/NOADAPTER (default)

Specifies the nexus number of the adapter to which the specified device is connected. It is a nonnegative 32-bit integer. The /NOADAPTER qualifier indicates that the device is not associated with any particular hardware. The /NOADAPTER qualifier is compatible with the /DRIVER_NAME qualifier only.

/CSR=csr-address

Specifies the CSR address for the device being configured. This address must be specified in hexadecimal. You must prefix the CSR address with %X. The CSR address is a quadword value that is loaded into IDB\$Q_CSR without any interpretation by SYSMAN. This address can be physical or virtual, depending on the specific device being connected:

- /CSR=%X3A0140120 for a physical address
- /CSR=%XFFFFFFFF807F8000 for a virtual address (the sign extension is required for AXP virtual addresses)

This qualifier is required if /ADAPTER=*tr-number* is specified.

/DRIVER_NAME=filespec

Specifies the name of the device driver to be loaded. If this qualifier is not specified, the default is obtained in the same manner as the SYSGEN default name. For example, if you want to load the SYS\$ELDRIVER.EXE supplied by Digital, the prefix SYS\$ must be present. Without the SYS\$, SYSMAN looks for ELDRIVER.EXE in SYS\$LOADABLE_IMAGES. This implementation separates the user device driver namespace from the device driver namespace supplied by Digital.

I/O Subsystem Configuration Commands in SYSMAN

IO CONNECT

/LOG=(ALL,CRB,DDB,DPT,IDB,SB,UCB)

/NOLOG (default)

Controls whether SYSMAN displays the addresses of the specified control blocks. The default value for the /LOG qualifier is /LOG=ALL. If /LOG=UCB is specified, a message similar to the following is displayed:

%SYSMAN-I-IOADDRESS, the UCB is located at address 805AB000

The default is /NOLOG.

/MAX_UNITS=maximum-number-of-units

Specifies the maximum number of units the driver can support. The default is specified in the driver prologue table (DPT) of the driver. If the number is not specified in the DPT, the default is 8. This number must be greater than or equal to the number of units specified by /NUM_UNITS. This qualifier is optional.

/NUM_UNITS=number-of-units

Specifies the number of units to be created. The starting device number is the number specified in the device name parameter. For example, the first device in DKA0 is 0. Subsequent devices are numbered sequentially. The default is 1. This qualifier is optional.

/NUM_VEC=vector-count

Specifies the number of vectors for this device. The default vector count is 1. The /NUM_VEC qualifier is optional. This qualifier should be used only when using the /VECTOR_SPACING qualifier. When using the /NUM_VEC qualifier, you must also use the /VECTOR qualifier to supply the base vector.

/SYS_ID=number-of-remote-system

Indicates the SCS system ID of the remote system to which the device is to be connected. It is a 64-bit integer; you must specify the remote system number in hexadecimal. The default is the local system. This qualifier is optional.

/VECTOR=(vector-address,...)

Specifies the interrupt vectors for the device or lowest vector. This is either a byte offset into the SCB of the interrupt vector for directly vectored interrupts or a byte offset into the ADP vector table for indirectly vectored interrupts. The values must be longword aligned. To specify the vector addresses in octal or hexadecimal, prefix the addresses with %O or %X, respectively. This qualifier is required when /ADAPTER=*tr-number* or /NUM_VEC=*vector-count* is specified. Up to 64 vectors can be listed.

/VECTOR_SPACING=number-of-bytes-between-vectors

Specifies the spacing between vectors. Specify the amount as a multiple of 16 bytes. The default is 16. You must specify both the base vector with /VECTOR and the number of vectors with /NUM_VEC. This qualifier is optional.

Description

The IO CONNECT command connects a hardware device and loads its driver, if the driver is not already loaded. You must have CMKRNL and SYSLCK privileges to use the IO CONNECT command.

I/O Subsystem Configuration Commands in SYSMAN IO CONNECT

Examples

1. SYSMAN> IO CONNECT DKA0:/DRIVER_NAME=SYSSDKDRIVER/CSR=%X80AD00-
/ADAPTER=4/NUM_VEC=3/VECTOR_SPACING=%X10/VECTOR=%XA20/LOG

%SYSMAN-I-IOADDRESS, the CRB is located at address 805AEC40
%SYSMAN-I-IOADDRESS, the DDB is located at address 805AA740
%SYSMAN-I-IOADDRESS, the DPT is located at address 80D2A000
%SYSMAN-I-IOADDRESS, the IDB is located at address 805AEE80
%SYSMAN-I-IOADDRESS, the SB is located at address 80417F80
%SYSMAN-I-IOADDRESS, the UCB is located at address 805B68C0

This command example connects device DKA0, loads driver SYSSDKDRIVER, and specifies the following:

- Physical CSR address
- Adapter number
- Number of vectors
- Spacing between vectors
- Interrupt vector address

The /LOG qualifier displays the addresses of all control blocks.

2. SYSMAN> IO CONNECT DKA0:/DRIVER_NAME=SYSSDKDRIVER/CSR=%X80AD00-
/ADAPTER=4/VECTOR=(%XA20,%XA30,%XA40)/LOG=(CRB,DPT,UCB)

%SYSMAN-I-IOADDRESS, the CRB is located at address 805AEC40
%SYSMAN-I-IOADDRESS, the DPT is located at address 80D2A000
%SYSMAN-I-IOADDRESS, the UCB is located at address 805B68C0

This command example connects device DKA0, loads driver SYSSDKDRIVER, and specifies the following:

- Physical CSR address
- Adapter number
- Addresses for interrupt vectors

The /LOG qualifier displays the addresses of the channel request block (CRB), the driver prologue table (DPT), and the unit control block (UCB).

3. SYSMAN> IO CONNECT FTA0:/DRIVER=SYSSFTDRIVER/NOADAPTER/LOG=(ALL)

%SYSMAN-I-IOADDRESS, the CRB is located at address 805AEC40
%SYSMAN-I-IOADDRESS, the DDB is located at address 805AA740
%SYSMAN-I-IOADDRESS, the DPT is located at address 80D2A000
%SYSMAN-I-IOADDRESS, the IDB is located at address 805AEE80
%SYSMAN-I-IOADDRESS, the SB is located at address 80417F80
%SYSMAN-I-IOADDRESS, the UCB is located at address 805B68C0

This command example connects pseudoterminal FTA0, loads driver SYSSFTDRIVER, and uses the /NOADAPTER qualifier to indicate that FTA0 is not an actual hardware device. The /LOG=ALL qualifier displays the addresses of all control blocks.

I/O Subsystem Configuration Commands in SYSMAN

IO LOAD

IO LOAD

Loads an I/O driver.

You must have CMKRNL and SYSLCK privileges to use the IO LOAD command.

Format

IO LOAD filespec

Parameters

filespec

Specifies the file name of the driver to be loaded. This parameter is required.

Qualifiers

/LOG=(ALL,DPT)

Controls whether SYSMAN displays information about drivers that have been loaded. The default value for the /LOG qualifier is /LOG=ALL. The driver prologue table (DPT) address is displayed when either /LOG=DPT or /LOG=ALL is specified.

Description

The IO LOAD command loads an I/O driver. You must have CMKRNL and SYSLCK privileges to use the IO LOAD command.

Example

```
SYSMAN> IO LOAD/LOG SYS$DKDRIVER
%SYSMAN-I-IOADDRESS, the DPT is located at address 80D5A000
```

This example loads device SYS\$DKDRIVER and displays the address of the driver prologue table (DPT).

IO SET PREFIX

Sets the prefix list that is used to manufacture the IOGEN Configuration Building Module (ICBM) names.

Format

IO SET PREFIX=(icbm-prefix)

Parameters

icbm-prefix

Specifies ICBM prefixes. These prefixes are used by the IO AUTOCONFIGURE command to build ICBM image names.

I/O Subsystem Configuration Commands in SYSMAN

IO SET PREFIX

Qualifiers

None.

Description

The IO SET PREFIX command sets the prefix list that is used to manufacture ICBM names.

Example

```
SYSMAN> IO SET PREFIX=(SYS$,PSI$,VME_)
```

This example specifies the prefix names used by IO AUTOCONFIGURE to build the ICBM names. The prefixes are SYS\$, PSI\$, and VME_.

IO SHOW BUS

On OpenVMS AXP systems, lists all the buses, node numbers, bus names, TR numbers, and base CSR addresses on the system. This display exists primarily for internal engineering support.

Format

```
IO SHOW BUS
```

Parameters

None.

Qualifiers

None.

Description

The IO SHOW BUS command lists all the buses, node numbers, bus names, TR numbers, and base CSR addresses. This display exists primarily for internal engineering support.

You must have CMKRNL privilege to use IO SHOW BUS.

I/O Subsystem Configuration Commands in SYSMAN

IO SHOW BUS

Example

```
SYSMAN> IO SHOW BUS
```

<u>_Bus</u>	<u>Node</u>	<u>TR#</u>	<u>Name</u>	<u>Base CSR</u>
LSB	0	1	EV3 4MB	FFFFFFFF86FA0000
LSB	6	1	MEM	FFFFFFFF86FC4000
LSB	7	1	MEM	FFFFFFFF86FCA000
LSB	8	1	IOP	FFFFFFFF86FD0000
XZA XMI-SCSI	0	3	XZA-SCSI	000008001880000
XZA XMI-SCSI	1	3	XZA-SCSI	000008001880000
XZA XMI-SCSI	0	4	XZA-SCSI	000008001900000
XZA XMI-SCSI	1	4	XZA-SCSI	000008001900000
XMI	4	2	LAMB	000008001A00000
DEMNA	0	5	Generic XMI	000008001E80000
DEMNA	0	6	Generic XMI	000008001F00000

This IO SHOW BUS example is from a DEC 7000 Model 600 AXP. Displays vary among different AXP systems. The indentation levels are deliberate in this display. They indicate the hierarchy of the adapter control blocks in the system. The column titles in the display have the following meanings:

Column Title	Meaning
Bus	Identity of the bus
Node	Index into the associated bus array (the bus slot)
TR#	Nexus number of the adapter to which the specified device is connected
Name	Name of the device
Base CSR	Base CSR address of the device

IO SHOW DEVICE

Displays information on device drivers loaded into the system, the devices connected to them, and their I/O databases. All addresses are in hexadecimal and are virtual.

Format

```
IO SHOW DEVICE
```

Parameters

None.

Qualifiers

None.

I/O Subsystem Configuration Commands in SYSMAN IO SHOW DEVICE

Description

The IO SHOW DEVICE command displays information on the device drivers loaded into the system, the devices connected to them, and their I/O databases.

The IO SHOW DEVICE command specifies that the following information be displayed about the specified device driver:

Driver	Name of the driver
Dev	Name of each device connected to the driver
DDB	Address of the device's device data block
CRB	Address of the device's channel request block
IDB	Address of the device's interrupt dispatch block
Unit	Number of each unit on the device
UCB	Address of each unit's unit control block

All addresses are in hexadecimal and are virtual.

Refer to the *OpenVMS System Manager's Manual* for additional information about SYSMAN.

Example

```
SYSMAN> IO SHOW DEVICE
```

The following is a sample display produced by the IO SHOW DEVICE command:

Driver	Dev	DDB	CRB	IDB	Unit	UCB
SYS\$FTDRIVER	FTA	802CE930	802D1250	802D04C0	0	801C3710
SYS\$EUDRIVER	EUA	802D0D80	802D1330	802D0D10	0	801E35A0
SYS\$DKDRIVER	DKI	802D0FB0	802D0F40	802D0E60	0	801E2520
SYS\$PKDRIVER	PKI	802D1100	802D13A0	802D1090	0	801E1210
SYS\$TTDRIVER						
OPERATOR						
NLDRIVER						

SYS\$TTDRIVER, OPERATOR, and NLDRIVER do not have devices associated with them.

I/O Subsystem Configuration Commands in SYSMAN

IO SHOW PREFIX

IO SHOW PREFIX

Displays the current prefix list used in the manufacture of IOGEN Configuration Building Module (ICBM) names.

Format

IO SHOW PREFIX

Parameters

None.

Qualifiers

None.

Description

The IO SHOW PREFIX command displays the current prefix list on the console. This list is used by the IO AUTOCONFIGURE command to build ICBM names.

Example

```
SYSMAN> IO SHOW PREFIX
%SYSMAN-I-IOPREFIX, the current prefix list is: SYS$,PSI$,VME_
```

This command example shows the prefixes used by IO AUTOCONFIGURE to build ICBM names.

Additional Considerations

In your role of supporting new users on OpenVMS AXP systems, you might encounter questions about the following additional topics:

- The Help Message utility. See Section B.1.
- The online Bookreader documentation provided on the OpenVMS AXP compact disc. See Section B.2.
- Unsupported DCL commands. See Section B.3.
- Differences in password generation display. See Section B.4.
- Default editor for the EDIT command is TPU. See Section B.5.
- The TECO editor. See Section B.6.
- Shareable images in the DEC C RTL. See Section B.7.
- Run-time libraries listed not included in this version of OpenVMS AXP. See Section B.8.
- Compatibility between the OpenVMS VAX and OpenVMS AXP Mathematics Libraries. See Section B.9.
- Linker utility enhancements. See Section B.10.

B.1 Help Message Utility

Help Message is a versatile utility that lets you quickly access online descriptions of system messages from the DCL prompt on a character-cell terminal (including DECterm windows).

Help Message displays message descriptions from the latest OpenVMS messages documentation (the most recent version of the *VMS System Messages and Recovery Procedures Reference Manual* plus any subsequent releases). In addition, the Help Message database can optionally include other source files, such as user-supplied messages documentation.

The staff of most medium-sized to large data centers often includes help desk personnel who answer questions about the computing environment from general users and programmers. Typically the system manager is a consultant or technical backup to the help desk specialists. If you find yourself in this role, you may want to alert the help desk personnel, as well as general users and programmers on OpenVMS AXP systems, about the availability of Help Message.

See the *OpenVMS System Messages: Companion Guide for Help Message Users* and the *OpenVMS System Manager's Manual* for details about using Help Message.

Additional Considerations

B.2 Online Documentation on Compact Disc

B.2 Online Documentation on Compact Disc

The OpenVMS Extended Documentation Set is included on the OpenVMS AXP Version 6.1 compact disc (CD) in DECW\$BOOK format. Users with a workstation and DECwindows Motif installed can view the manuals with the Bookreader application. Refer to the *OpenVMS AXP Version 6.1 CD-ROM User's Guide* for a list of the manuals on the CD and information about enabling access to and reading the online documents.

B.3 Unsupported DCL Commands

The following DCL commands are not supported on OpenVMS AXP:

- MONITOR POOL
- SET FILE/UNLOCK
- UNLOCK

B.4 Password Generation

On OpenVMS AXP systems, the password generation algorithm allows for future use of generation databases for non-English passwords. Because of this, the password generation logic does not perform English word hyphenation. As a result, the SET PASSWORD command cannot display a hyphenated word list as it does on OpenVMS VAX systems. This change is permanent on OpenVMS AXP and is intended to accommodate possible future support for alternate-language password generation databases.

B.5 Default Editor for EDIT Command

The default editor for the EDIT command on OpenVMS AXP (and on OpenVMS VAX V6.*n*) is TPU. The default editor on VAX VMS Version 5.*n* and earlier releases is EDT. When users enter the EDIT command, the Extensible Versatile Editor (EVE) is invoked rather than the EDT editor. The EDT editor is still included with OpenVMS AXP.

If your users prefer to continue using the EDT editor, have them define the following symbol interactively or in their login command procedure:

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

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

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

Note that the default editor for the Mail utility (MAIL) also has been changed to the DECTPU-based EVE editor rather than EDT. By entering the MAIL command SET EDITOR, you can specify that a different editor be invoked instead of the DECTPU editor. For example, to select the EDT editor, issue the MAIL command SET EDITOR EDT. The EDT editor remains your default MAIL editor (even if you log out of the system and log back in) until you enter the SET EDITOR TPU command.

Additional Considerations

B.5 Default Editor for EDIT Command

Users also can define the logical name MAIL\$EDIT to be a command file before entering MAIL. When they issue any MAIL command that invokes an editor, the command file will be called to perform the edit. In the command file, you can also invoke other utilities, such as the spellchecker, and you can specify any function that can be done in a command file.

If desired, another option is to override the selected MAIL editor temporarily by defining MAIL\$EDIT to be the string CALLABLE_ with the name of the desired editor appended. For example, to use callable EDT rather than callable EVE, users can type the following command:

```
$ DEFINE MAIL$EDIT CALLABLE_EDT
```

In EVE, you can select an EDT-like keypad by defining the OpenVMS AXP logical name EVE\$KEYPAD to be EDT. See the *OpenVMS AXP Version 6.1 Release Notes* for details about how to do this at either the process level or the system level. You can find an example of how to define EVE\$KEYPAD at the system level in the file SYSS\$STARTUP:SYLOGICALS.TEMPLATE.

The general release notes chapter of the *OpenVMS AXP Version 6.1 Release Notes* also contains a complete description of the EDIT command, DECTPU, and EVE.

See the *Guide to the DEC Text Processing Utility* and the *OpenVMS User's Manual* for more information about DECTPU and EVE.

B.6 TECO Editor

The TECO editor is included in OpenVMS AXP. Invoke the TECO editor with the EDIT/TECO command as described in the *OpenVMS DCL Dictionary* or with more traditional access methods. For information about the use of TECO, see the *PDP-11 TECO User's Guide*.

B.7 Shareable Images in the DEC C RTL for OpenVMS AXP

If you answer problem reports submitted by programmers who are coding C applications on OpenVMS AXP systems, you might receive questions about the shareable images in the DEC C Run-Time Library (RTL).

On AXP systems, the DEC C RTL does not provide the VAXCRTL.EXE or VAXCRTL.G.EXE shareable images. Instead, the image DECC\$SHR.EXE (which resides in IMAGELIB) must be used. This image contains all DEC C RTL functions and data and has an OpenVMS conformant namespace (all external names are prefixed with DECC\$). To use this image, all DEC C RTL references must be prefixed with DECC\$ so that the proper code in DECC\$SHR.EXE is accessed.

On an AXP system that has DEC C installed, type "HELP CC /PREFIX" at the DCL prompt for a description of the prefixing behavior of the compiler. To resolve nonprefixed names, programmers can link against object libraries SYSS\$LIBRARY:VAXCRTL.OLB, SYSS\$LIBRARY:VAXCRTL.D.OLB, SYSS\$LIBRARY:VAXCRTL.T.OLB, and SYSS\$LIBRARY:VAXCCURSE.OLB. On VAX systems, those libraries contain object code for the RTL support; on AXP systems, those libraries contain jacket routines to the prefixed entry points.

Note that on VAX systems, you use an option file when using VAXCRTL.EXE. On AXP systems, you do not use an option file when using DECC\$SHR.EXE because it is in SYSS\$LIBRARY:IMAGELIB.OLB.

Additional Considerations

B.7 Shareable Images in the DEC C RTL for OpenVMS AXP

See the *DEC C Run-Time Library Reference Manual for OpenVMS Systems* and the *DEC C Reference Supplement for OpenVMS AXP Systems* for more information.

B.8 Run-Time Libraries

Table B-1 lists the run-time libraries that are not included in OpenVMS AXP Version 6.1.

Table B-1 Run-Time Libraries Not Included in OpenVMS AXP Version 6.1

DBGSSISHR	DEBUG item is not required on OpenVMS AXP.
DNSSRTL, DNS\$SHARE, and DTI\$SHARE	DNS is not supported.
EPM\$SRVSHR	DETrace is not supported.
VBLAS1RTL	OpenVMS VAX vector programs are not supported.
VMTHRTL	OpenVMS VAX vector programs are not supported.

Most run-time libraries that are available in OpenVMS VAX are available in OpenVMS AXP Version 6.1. The OpenVMS VAX libraries that are not available are either not being ported to OpenVMS AXP or are planned for a later release of OpenVMS AXP.

For example, the vector math libraries VBLAS1RTL and VMTHRTL are not available in OpenVMS AXP because there is no support on OpenVMS AXP for programs that use the OpenVMS VAX vector instructions.

B.9 Compatibility Between the OpenVMS VAX and OpenVMS AXP Mathematics Libraries

Mathematical applications using the standard OpenVMS call interface to the OpenVMS Run-Time Mathematics (MTH\$) Library need not change their calls to MTH\$ routines when migrating to an 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. However, there is no support in the DPML for calls made to JSB entry points and vector routines. Please note that DPML routines are different from those in the OpenVMS Run-Time Mathematics (MTH\$) Library. You should expect to see small differences in the precision of the mathematical results.

If one of your goals is to maintain compatibility with future libraries and to create portable mathematical applications, Digital recommends that you use the DPML routines available through the high-level language of your choice (for example, Fortran and C) rather than using the call interface. Significantly higher performance and accuracy are also available with DPML routines.

See the *Digital Portable Mathematics Library* manual for more information about DPML.

B.10 Linker Utility Enhancements

The following sections describe new AXP enhancements to the OpenVMS Linker utility. Refer to the online version of the *OpenVMS Linker Utility Manual* for additional information on these enhancements.

Additional Considerations

B.10 Linker Utility Enhancements

B.10.1 New /DSF Qualifier

The /DSF qualifier directs the linker to create a file called a debug symbol file (DSF) for use by the OpenVMS AXP System-Code Debugger. The default is /NODSF. The /DSF qualifier can be used with the /NOTRACEBACK qualifier to suppress the appearance of SYSS\$IMGSTA in the image's transfer array. The /DSF qualifier has no effect on the contents of the image, including the image header.

The /DSF and /DEBUG qualifiers are not mutually exclusive. However, the combination is not generally useful. The debug bit in the image header will be set and SYSS\$IMGSTA will be included in the transfer array, but there will be no information for the symbolic debugger included in the image. The DSF file will be generated as usual.

Use the following format:

```
LINK/DSF[=file-spec]
```

See *OpenVMS AXP Device Support: Developer's Guide* for guidelines on using the OpenVMS AXP System-Code Debugger.

B.10.2 New /ATTRIBUTES Qualifier for COLLECT= Option

The COLLECT= option directs the linker to place the specified program section (or program sections) into the specified cluster. A new qualifier, /ATTRIBUTES, directs the linker to mark the cluster *cluster-name* with the indicated qualifier keyword value. This qualifier is used to build AXP drivers.

Use the following format:

```
COLLECT=cluster-name [/ATTRIBUTES={RESIDENT | INITIALIZATION_CODE}],-  
psect-name[,...])
```

Qualifier Values

RESIDENT

Marks the cluster *cluster-name* as RESIDENT so that the image section created from that cluster has the EISDSV_RESIDENT flag set. This will cause the loader to map the image section into nonpaged memory.

INITIALIZATION_CODE

Marks the cluster 'cluster-name' as INITIALIZATION_CODE so that the image section created from that cluster has the EISDSV_INITALCOD flag set. The initialization code will be executed by the loader. This keyword is specifically intended for use with program sections from modules SYSS\$DOINIT and SYSS\$DRIVER_INIT in STARLET.OLB.

See *OpenVMS AXP Device Support: Developer's Guide* for guidelines on using this qualifier.

A

ACP_DINDXCACHE system parameter
 unit change on AXP, 5-2

ACP_DIRCACHE system parameter
 unit change on AXP, 5-2

ACP_HDRCACHE system parameter
 unit change on AXP, 5-2

ACP_MAPCACHE system parameter
 unit change on AXP, 5-2

ACP_WORKSET system parameter
 unit change on AXP, 5-2

Adaptive pool management, 5-11

Aliases
 See Cluster aliases

ALLOCATE command, 3-1

ANALYZE/AUDIT command, 3-1

ANALYZE/CRASH_DUMP command, 3-1

ANALYZE/DISK_STRUCTURE command, 3-1

ANALYZE/ERROR_LOG command
 restrictions on AXP, 3-1

ANALYZE/IMAGE command, 3-1

ANALYZE/MEDIA command, 3-1

ANALYZE/OBJECT command, 3-1

ANALYZE/PROCESS_DUMP command, 3-1

ANALYZE/RMS_FILE command, 3-1

ANALYZE/SYSTEM command, 3-1

Architectures
 determining type using F\$GETSYI lexical
 function, 2-7

ARCH_NAME argument
 to F\$GETSYI lexical function, 2-8

ARCH_TYPE argument
 to F\$GETSYI lexical function, 2-8

/ATTRIBUTES qualifier
 in Linker utility, B-5

Authorize utility (AUTHORIZE)
 commands and parameters, 2-5

Autoconfiguration
 on AXP, A-1

AUTOCONFIGURE command
 See IO AUTOCONFIGURE command

AUTOGEN.COM command procedure
 running, 2-26

AWSMIN system parameter
 changed default value on AXP, 5-6
 dual values on AXP, 5-4

AXP systems
 DCL commands unsupported
 See DCL
 default editor, B-2
 differences in system management from VAX
 overview, 1-2
 OpenVMS binaries
 on compact disc, 2-24
 OpenVMS documentation
 on compact disc, 2-24, B-2
 page size, 1-3
 pagelet size, 1-3
 performance
 adaptive pool management, 5-11
 improving for images, 5-13
 system parameter measurement changes
 for larger page size, 5-1
 use of page or pagelet values by utilities
 and commands, 5-7
 virtual I/O cache, 5-17
 run-time libraries, B-4
 symmetric multiprocessing configurations,
 2-18

B

Backups
 maintenance tasks, 3-1
 of AXP data on AXP computers, 2-4
 on AXP and restore on VAX, 3-1
 on VAX and restore on AXP, 3-1

BALSETCNT system parameter
 changed default value on AXP, 5-6

Batch and print queuing system
 comparison of AXP and VAX capabilities, 3-4

Bookreader application
 versions of OpenVMS AXP documentation, B-2

BOOT command
 boot flags, 2-10
 characteristics and use on AXP, 2-10
 console subsystem actions in response to, 2-11
 qualifiers and parameters, 2-10

Boot flags
 DBG_INIT, 2-11
 USER_MSGS, 2-11
BORROWLIM system parameter
 units that remain as CPU-specific pages on
 AXP, 5-3
Buses
 display on AXP, A-7

C

C programming language
 run-time library
 using DECCSSHR, B-3
 using DPML routines, B-4
Caches
 virtual I/O
 observing statistics, 5-17
CD-ROM (compact disc read-only memory)
 on OpenVMS AXP
 documentation, 2-24, B-2
 installation media, 2-24
CI
 not supported for DECnet on AXP, 6-4
CLEAR LINE command, 6-6
CLEAR MODULE X25-ACCESS command, 6-6
CLEAR MODULE X25-PROTOCOL command,
 6-6
CLEAR MODULE X25-SERVER command, 6-6
CLEAR MODULE X29-SERVER command, 6-6
CLEAR NODE command, 6-6
CLISYMTBL system parameter
 changed default value on AXP, 5-6
 unit change on AXP, 5-2
CLUE (Crash Log Utility Extractor)
 See Crash Log Utility Extractor
 CLEANUP command, 3-9
 CONFIG command, 3-9
 CRASH command, 3-9
 DEBUG command, 3-9
 HISTORY command, 3-9
 MCHK command, 3-9
 MEMORY command, 3-9
 PROCESS command, 3-9
 STACK command, 3-9
 VCC command, 3-9
 XQP command, 3-9
CLUE(Crash Log Utility Extractor)
 ERRLOG command, 3-9
Cluster aliases
 NCP commands to enable, 6-3
 supported with DECnet on AXP, 6-1
Computer interconnect
 See CI
Configuring the DECnet database, 6-2

Configuring the I/O subsystem
 on AXP, 2-16
CONNECT command
 See IO CONNECT command
Connecting hardware devices, A-3
CONSCOPY command procedure
 not available on AXP, 2-12
Conserving dump file storage space, 3-8
Console subsystem
 actions in response to BOOT command, 2-11
Console volumes
 copying
 CONSCOPY.COM not available on AXP,
 2-12
 CONVERT/RECLAIM command, 3-2
 CONVSHR shareable library, 3-2
CPU-specific pages
 See Pages
Crash Log Utility Extractor (CLUE)
 commands
 archiving information, 3-9
 CSR addresses
 display on AXP, A-7
 CTLIMGLIM system parameter
 unit change on AXP, 5-2
 CTLPAGES system parameter
 unit change on AXP, 5-2
Ctrl/T key sequence
 displays AXP CPU-specific page values, 5-9

D

DBG_INIT boot flag, 2-11
DCL (DIGITAL Command Language)
 unsupported commands, B-2
 with AXP CPU-specific page values, 5-7
 with AXP pagelet values, 5-7
DDCMP
 not supported on AXP, 6-4
Debug symbol file
 See also /DSF qualifier
 creating, B-5
DEC 7000 Model 600 AXP
 DSA local device naming, 2-23
DEC C
 See C programming language
DEC File Optimizer for OpenVMS
 defragmenting disks, 3-3
DEC InfoServer
 See InfoServer
DECdtm services
 related MONITOR TRANSACTION command
 supported on AXP, 3-2
 supported on AXP, 2-5
DECEvent utility
 description, 3-4
 report types, 3-4

- DECnet
 - cluster alias supported, 6-1
 - DECNET object, 6-2
 - migration issues, 6-1
 - P.S.I. not supported on AXP, 6-4
 - PAK name difference using DECnet for OpenVMS AXP, 2-14
- DECnet-VAX
 - See DECnet
 - See Network management tasks
- Decompressing libraries, 2-1
- DECW\$TAILOR utility
 - See DECwindows
- DECwindows
 - Tailoring utility (DECW\$TAILOR) supported on AXP, 2-5
- DEFINE EXECUTOR command, 6-7
- Defragmenting disks, 3-3
- Device drivers
 - configuring, A-8
 - showing information, A-8
 - Step 1, 2-3
 - Step 2, 2-3
 - user written, 2-3
 - written in C, 2-3
- Device naming
 - DSA
 - differences on AXP and VAX, 2-23
- Devices
 - on AXP computers, 2-22
- DIGITAL Command Language
 - See DCL
- Digital data communications message protocol
 - See DDCMP
- DIGITAL Storage Architecture disks
 - See DSA disks
- DISABLE AUTOSTART/QUEUES command
 - /ON_NODE qualifier, 3-5
- Disks
 - defragmenting
 - movefile** subfunction supported on AXP and VAX, 3-3
 - DSA local device naming, 2-23
- Distributed Name Service
 - See DNS node name interface
- DNS node name interface
 - interface option not supported on AXP, 6-7
 - not supported by DECnet, 6-6
 - not supported on AXP, 6-4
- Documentation
 - on OpenVMS AXP compact disc, 2-24, B-2
- Documentation comments, sending to Digital, iii
- Downline loading, 6-2

- DPML (Digital Portable Mathematics Library), B-4
- Drivers
 - supplied by Digital
 - file name format on AXP, 2-24
- DSA disks
 - local device naming
 - differences on AXP and VAX, 2-23
- /DSF qualifier, B-5
- DTS/DTR (DECnet Test Sender/DECnet Test Receiver utility), 6-2
- Dual values for system parameters
 - on AXP, 5-3
- Dump file information
 - saving automatically, 3-9
- Dump files
 - conserving storage space, 3-8
- DUMPSTYLE system parameter
 - changed default value on AXP, 5-7
 - controlling size of system dump files, 3-8
- DVNETEND PAK
 - DECnet end node license, 6-1
- DVNETEXT PAK
 - DECnet for OpenVMS AXP extended license, 6-1
- DVNETRTG PAK
 - DECnet for OpenVMS VAX routing license, 6-1

E

- EDIT command
 - default editor changed to TPU, B-2
 - EDT, B-2
 - selecting EDT keypad in EVE, B-3
 - used with DCL command procedures, B-2
 - overriding the default editor, B-2
 - /TECO, B-3
- Editors
 - on AXP
 - default, B-2
- ENABLE AUTOSTART/QUEUES command
 - /ON_NODE qualifier, 3-5
- End-node support
 - on AXP, 6-3
- ERLBUFFERPAGES system parameter
 - unit change on AXP, 5-2
- Ethernet monitor (NICONFIG), 6-2
- Event logging, 6-2
- Executable files (.EXE)
 - planning for and managing location, 2-7
- Executive images
 - improving the performance of
 - using GHRs, 5-15
 - SYS.EXE on VAX renamed to SYSSBASE_IMAGE.EXE on AXP, 2-26
- External values
 - pagelet units for some system parameters
 - on AXP, 5-3

F

- F\$GETQUI lexical function, 3–6
- F\$GETSYI lexical function
 - ARCH_NAME argument, 2–7
 - ARCH_TYPE argument, 2–7
 - HW_NAME argument, 2–7
 - NODENAME argument, 2–7
 - PAGE_SIZE argument, 2–7
- FAL (file access listener)
 - on AXP, 6–2
- FDDI (Fiber Distributed Data Interface)
 - support on AXP, 6–2
- Feedback on documentation, sending to Digital, iii
- Fiber Distributed Data Interface
 - See FDDI
- Fiber-optic cables
 - See FDDI
- File access listener
 - See FAL
- File systems
 - similarities on AXP and VAX, 3–1
- File transfers
 - with FAL, 6–2
- Fortran
 - using DPML routines, B–4
- FREEGOAL system parameter
 - units that remain as CPU-specific pages on AXP, 5–3
- FREELIM system parameter
 - units that remain as CPU-specific pages on AXP, 5–3
- Full-checking multiprocessing synchronization images, 2–18

G

- GBLPAGES system parameter
 - changed default value on AXP, 5–6
 - dual values on AXP, 5–3
- GBLPAGFIL system parameter
 - changed default value on AXP, 5–6
 - units that remain as CPU-specific pages on AXP, 5–3
- \$GETQUI system service, 3–6
- GHRs (granularity hint regions)
 - improving performance of images, 5–13
 - SHOW MEMORY/GH_REGIONS command, 5–15
- GH_EXEC_CODE system parameter, 5–14
- GH_EXEC_DATA system parameter, 5–14
- GH_RES_CODE system parameter, 5–14
- GH_RES_DATA system parameter, 5–14
- GH_RSRVPGCNT system parameter, 5–14

Granularity hint regions

See GHRs

- GROWLIM system parameter
 - units that remain as CPU-specific pages on AXP, 5–3

H

- Hardware types
 - determining type using F\$GETSYI lexical function, 2–7
- HELP command
 - /MESSAGE qualifier, B–1
- Help Message utility, B–1

I

- I/O configuration support in SYSMAN, A–1
- I/O databases, A–8
- I/O subsystem configuration
 - comparison of I/O subsystem commands on AXP and VAX, 2–16
 - using SYSMAN instead of SYSGEN on AXP, 2–16
- Images
 - improving performance, 2–26, 5–13
 - Multiprocessing synchronization, 2–18
 - patching
 - not supported on AXP, 3–10
- IMGIOCNT system parameter
 - unit change on AXP, 5–2
- Improving performance of images, 2–26
- InfoServer
 - supported on AXP, 2–4
- INITIALIZE command
 - /QUEUE qualifier
 - qualifiers using AXP pagelet values, 5–11
- INITIALIZE/QUEUE command
 - /AUTOSTART_ON qualifier, 3–5
 - comparison on AXP and VAX, 3–5
- Install utility (INSTALL)
 - GHR feature on AXP
 - /RESIDENT qualifier, 5–14
 - improving performance of images, 2–26
 - /RESIDENT qualifier to ADD, CREATE, REPLACE, 5–14
- Installations
 - comparison between VMSINSTAL and POLYCENTER Software Installation utility, 2–7
 - media on OpenVMS AXP
 - compact disc, 2–24
- Installed products
 - on AXP
 - listing, 2–25
- Installed resident images
 - Install utility support, 5–14
 - SYSGEN support, 5–14

INSTALLED_PRDS.COM command procedure
 listing installed products, 2-25
 Internal values
 CPU-specific page units for some system
 parameters
 on AXP, 5-3
 Invoking SDA by default, 3-9
 IO AUTOCONFIGURE command
 in SYSMAN, A-1
 IO CONNECT command
 in SYSMAN, A-3
 IO LOAD command
 in SYSMAN, A-6
 IO SET PREFIX command
 in SYSMAN, A-6
 IO SHOW BUS command
 in SYSMAN, A-7
 IO SHOW DEVICE command
 in SYSMAN, A-8
 IO SHOW PREFIX command
 in SYSMAN, A-10

L

LAD Control Program (LADCP) utility
 supported on AXP, 2-5
 LASTport network transport control program
 supported on AXP, 2-5
 LAT Control Program (LATCP) utility, 2-4
 LAT startup, 2-4
 LATACP (LAT ancillary control process), 2-4
 LATSYP symbiont, 2-4
 Level 1 routers
 supported on AXP for cluster alias only, 6-3
 Level 2 routers
 not supported on AXP, 6-3
 License Management Facility
 See LMF
 Lines
 X.25 testing
 not supported on AXP, 6-6
 LINK command
 /SECTION_BINDING qualifier
 GHR feature, 5-13
 Linker utility
 enhancements, B-5
 LMF (License Management Facility)
 on OpenVMS AXP, 2-13
 LNMPHASHTBL system parameter
 changed default value on AXP, 5-6
 LNMSHASHTBL system parameter
 changed default value on AXP, 5-6
 LOAD command
 See IO LOAD command
 Loader changes, 5-15

Loading I/O drivers, A-6
 Local Area Disk Control Program
 See LAD Control Program (LADCP) utility
 Local Area Transport Control Program
 See LAT Control Program (LATCP) utility
 Logging events, 6-2
 LOOP LINE command
 not supported on AXP, 6-6
 Loopback mirror testing, 6-2
 LTPAD process, 2-4

M

Mail utility (MAIL)
 change in default editor, B-2
 choosing an editor, B-2
 editing mail using a command file, B-3
 MAIL\$EDIT logical, B-3
 Maintenance tasks
 comparison of batch and print queuing systems
 on AXP and VAX, 3-4
 differences on AXP and VAX, 3-3
 MONITOR POOL not provided, 3-2
 MONITOR VECTOR, 3-2
 Patch utility not supported, 3-10
 size of system dump files on AXP, 3-7
 similarities on AXP and VAX, 3-1
 Accounting utility commands, 3-1
 ALLOCATE command, 3-1
 ANALYZE/AUDIT, 3-1
 ANALYZE/CRASH_DUMP, 3-1
 ANALYZE/DISK_STRUCTURE, 3-1
 ANALYZE/IMAGE command, 3-1
 ANALYZE/MEDIA, 3-1
 ANALYZE/OBJECT command, 3-1
 ANALYZE/PROCESS_DUMP, 3-1
 ANALYZE/RMS_FILE, 3-1
 ANALYZE/SYSTEM, 3-1
 analyzing error logs on AXP, 3-1
 backup of AXP data on AXP systems, 3-1
 CONVERT/RECLAIM, 3-2
 CONVSHR library, 3-2
 defragmenting disks, 3-3
 file system, 3-1
 MONITOR ALL_CLASSES, 3-2
 MONITOR CLUSTER, 3-2
 MONITOR DECNET, 3-2
 MONITOR DISK, 3-2
 MONITOR DLOCK, 3-2
 MONITOR FCP, 3-2
 MONITOR FILE_SYSTEM_CACHE, 3-2
 MONITOR IO, 3-2
 MONITOR LOCK, 3-2
 MONITOR MODES (with exceptions), 3-2
 MONITOR PAGE, 3-2
 MONITOR PROCESSES, 3-2
 MONITOR RMS, 3-2

Maintenance tasks

- similarities on AXP and VAX (cont'd)
 - MONITOR STATES, 3-2
 - MONITOR SYSTEM, 3-2
 - MONITOR TRANSACTION, 3-2
 - MONITOR VECTOR (with exceptions), 3-2
 - MONITOR/INPUT, 3-2
 - MONITOR/RECORD, 3-2
 - MOUNT command, 3-1
 - SUMSLP utility, 3-3
 - SYSMAN utility, 3-3
- Mathematics run-time library
 - on AXP, B-4
- MAXBUF system parameter
 - changed default value on AXP, 5-6
- Maximum network size
 - comparison on AXP and VAX, 6-2
- MINWSCNT system parameter
 - unit change on AXP, 5-2
- MONITOR ALL_CLASSES command, 3-2
- MONITOR CLUSTER command, 3-2
- MONITOR DECNET command, 3-2
- MONITOR DISK command, 3-2
- MONITOR DLOCK command, 3-2
- MONITOR FCP command, 3-2
- MONITOR FILE_SYSTEM_CACHE command, 3-2
- MONITOR /INPUT command, 3-2
- MONITOR IO command, 3-2
- MONITOR LOCK command, 3-2
- MONITOR MODES command, 3-2
- MONITOR PAGE command, 3-2
- MONITOR POOL command
 - not provided, 3-2, B-2
- MONITOR PROCESSES command, 3-2
 - displays AXP CPU-specific page values, 5-9
- MONITOR /RECORD command, 3-2
- MONITOR RMS command, 3-2
- MONITOR STATES command, 3-2
- MONITOR SYSTEM command, 3-2
- MONITOR TRANSACTION command, 3-2
- MONITOR VECTOR command, 3-2
- MOUNT command, 3-1
- Movefile subfunction
 - supported on AXP and VAX, 3-3
- MPW_HILIMIT system parameter
 - changed default value on AXP, 5-6
 - units that remain as CPU-specific pages on AXP, 5-3
- MPW_LOLIMIT system parameter
 - changed default value on AXP, 5-6
 - units that remain as CPU-specific pages on AXP, 5-3
- MPW_LOWAITLIMIT system parameter
 - changed default value on AXP, 5-6
 - units that remain as CPU-specific pages on AXP, 5-3

- MPW_THRESH system parameter
 - changed default value on AXP, 5-6
 - units that remain as CPU-specific pages on AXP, 5-3
- MPW_WAITLIMIT system parameter
 - changed default value on AXP, 5-6
 - units that remain as CPU-specific pages on AXP, 5-3
- MPW_WRTCLUSTER system parameter
 - changed default value on AXP, 5-6
 - units that remain as CPU-specific pages on AXP, 5-3
- Multiprocessing synchronization images, 2-18
- MULTIPROCESSING system parameter
 - on AXP and VAX, 2-15

N

- NCP (Network Control Program)
 - command parameters affected by unsupported features, 6-5
 - enabling cluster alias, 6-3
- NETCONFIG.COM command procedure, 6-2
- NETCONFIG_UPDATE.COM command procedure, 6-2
- Network access
 - starting, 6-2
- Network Control Program utility
 - See NCP
- Network management tasks
 - differences on AXP and VAX, 6-3
 - CI not supported for DECnet on AXP, 6-4
 - DDCMP not supported on AXP, 6-4, 6-5
 - DNS node name interface not supported on AXP, 6-4
 - level 2 routing not supported on AXP, 6-5
 - NCP command parameters affected by unsupported features, 6-5
 - routing not supported on AXP, 6-3
 - routing support, 6-2
 - VAX P.S.I. not supported on AXP, 6-4, 6-5
 - X.25 circuits on AXP, 6-5
 - similarities on AXP and VAX
 - configuring the DECnet database, 6-2
 - DECnet cluster alias, 6-1
 - DECnet objects and associated accounts, 6-2
 - downline loading, 6-2
 - DTS/DTR, 6-2
 - DVNETEND end node license, 6-1
 - end-node support on AXP, 6-3
 - end-node support/nomaster, 6-2
 - Ethernet monitor (NICONFIG), 6-2
 - Ethernet support, 6-2
 - event logging, 6-2
 - FDDI support, 6-2
 - file access listener, 6-2
 - file transfer, 6-2

Network management tasks
similarities on AXP and VAX (cont'd)
 loopback mirror testing, 6-2
 maximum network size, 6-2
 NETCONFIG.COM, 6-2
 NETCONFIG_UPDATE.COM, 6-2
 node name rules, 6-2
 SET HOST capabilities, 6-2
 starting network access, 6-2
 STARTNET.COM, 6-2
 task-to-task communication, 6-2
 upline dump, 6-2

Networking

See also Network management tasks

Networks

management
 migration issues, 6-1

NICONFIG (Ethernet Configurator), 6-2

Node numbers

display on AXP, A-7

NPAGEDYN system parameter

changed default value on AXP, 5-6

O

Object files

planning for and managing location, 2-7

OPCOM process, 2-5

OpenVMS AXP

See AXP systems

P

P.S.I.

See VAX P.S.I.

not supported on AXP, 6-4

PAGEDYN system parameter

changed default value on AXP, 5-6

Pagelets

size, 1-3

 impact on tuning, 5-1

 system parameters changed in unit name only
 from VAX to AXP, 5-2

 system parameters with dual values on AXP,
 5-3

 utilities and commands qualifiers using pagelet
 values, 5-7

Pages

determining size using F\$GETSYI lexical
function, 2-7

size, 1-3

 impact on tuning, 5-1

 system parameters changed in unit name only
 from VAX to AXP, 5-2

 system parameters that remain as CPU-specific
 units on AXP, 5-2

 system parameters with dual values on AXP,
 5-3

Pages (cont'd)

 utilities and commands qualifiers using page
 values, 5-7

PAGE_SIZE argument

 to F\$GETSYI lexical function, 2-8

PAGTBLPFC system parameter

 dual values on AXP, 5-3

PAKs (Product Authorization Keys)

 DVNETEND DECnet end node license on AXP
 and VAX, 6-1

 DVNETEXT DECnet for OpenVMS AXP
 extended license, 6-1

 DVNETRTG DECnet for OpenVMS VAX routing
 license, 6-1

Passwords

 generated by system

 differences on AXP and VAX, B-2

Patch utility (PATCH)

 not supported on AXP, 3-10

Performance optimization tasks

AXP images

 installing resident, 2-26, 5-13

 differences on AXP and VAX, 5-1

 adaptive pool management, 5-11

 impact of larger page size on tuning, 5-1

 improving performance of images, 5-13

 tuning impact of larger page size, 5-1

 virtual I/O cache, 5-17

PFCDEFAULT system parameter

 dual values on AXP, 5-3

PHYSICALPAGES system parameter, 2-15

PHYSICAL_MEMORY system parameter, 2-15

PIOPAGES system parameter

 unit change on AXP, 5-2

POLYCENTER Software Installation utility

 comparison with VMSINSTAL, 2-7

Pool management

 See also Adaptive pool management

POOLCHECK system parameter, 2-15

PQL_DBIOLM system parameter

 changed default value on AXP, 5-6

PQL_DBYTLM system parameter

 changed default value on AXP, 5-6

PQL_DDIOLM system parameter

 changed default value on AXP, 5-7

PQL_DENQLM system parameter

 changed default value on AXP, 5-7

PQL_DFILLM system parameter

 changed default value on AXP, 5-7

PQL_DPGFLQUOTA system parameter

 changed default value on AXP, 5-7

 dual values on AXP, 5-4

PQL_DPRCLM system parameter

 changed default value on AXP, 5-7

PQL_DTQELM system parameter

 changed default value on AXP, 5-7

PQL_DWSDEFAULT system parameter
 changed default value on AXP, 5-7
 dual values on AXP, 5-4

PQL_DWSEXTENT system parameter
 changed default value on AXP, 5-7
 dual values on AXP, 5-4

PQL_DWSQUOTA system parameter
 changed default value on AXP, 5-7
 dual values on AXP, 5-4

PQL_MPGFLQUOTA system parameter
 changed default value on AXP, 5-7
 dual values on AXP, 5-4

PQL_MWSDEFAULT system parameter
 changed default value on AXP, 5-7
 dual values on AXP, 5-4

PQL_MWSEXTENT system parameter
 changed default value on AXP, 5-7
 dual values on AXP, 5-4

PQL_MWSQUOTA system parameter
 changed default value on AXP, 5-7
 dual values on AXP, 5-4

PRINT command
 /RETAIN qualifier, 3-5

Print queuing system
 See Batch and print queuing system

Product installation log file
 on AXP, 2-25

PURGE LINE command, 6-6

PURGE MODULE X25-ACCESS command, 6-6

PURGE MODULE X25-PROTOCOL command,
 6-6

PURGE MODULE X25-SERVER command, 6-6

PURGE MODULE X29-SERVER command, 6-6

PURGE NODE command, 6-6

R

RMS Journaling
 similarities on AXP and VAX, 2-3

Rounding-up algorithm
 input pagelet values to whole pages
 on AXP, 2-27

Routing
 not supported on AXP, 6-3

RSRVPAGCNT system parameter
 units that remain as CPU-specific pages on
 AXP, 5-3

RUN command
 process
 qualifiers using AXP pagelet values, 5-11

Run-time libraries
 on AXP, B-4

S

SDA (System Dump Analyzer utility)
 See System Dump Analyzer utility

SDA CLUE commands
 collecting dump file information, 3-9

Security
 audit log file
 SECURITY.AUDIT\$JOURNAL, 2-5
 SECURITY_AUDIT.AUDIT\$JOURNAL,
 2-5

Security tasks, 4-1

SECURITY.AUDIT\$JOURNAL audit log file, 2-5

SECURITY_AUDIT.AUDIT\$JOURNAL audit log
 file, 2-5

SET ENTRY command
 comparison on AXP and VAX, 3-6
 qualifiers using AXP pagelet values, 5-11
 /RETAIN qualifier, 3-5

SET EXECUTOR command, 6-7

SET FILE/UNLOCK command
 not supported on AXP, B-2

SET HOST command
 on AXP, 6-2

SET PASSWORD command
 different display, B-2

SET PREFIX command
 See IO SET PREFIX command

SET QUEUE command
 qualifiers using AXP pagelet values, 5-11

SET SECURITY command
 /ACL qualifier, 3-6
 /CLASS qualifier, 3-6

SET WORKING_SET command
 /QUOTA qualifier
 specifying AXP pagelet values, 5-10

Setup tasks, 2-1
 differences on AXP and VAX, 2-5

AUTOGEN, 2-26

CONSCOPY.COM not available on AXP,
 2-12

devices, 2-22

DSA device naming, 2-23

file name format of drivers supplied by
 Digital, 2-24

I/O subsystem configuration commands in
 SYSMAN utility on AXP, 2-15

improving the performance of images, 2-26

installation media, 2-24

new and changed parameters, 2-15

startup command procedures, 2-22

SYSMAN utility, 2-16

Terminal Fallback Facility, 2-27

VMSINSTAL utility, 2-24

similarities on AXP and VAX, 2-1

Authorize utility commands and
 parameters, 2-5

Setup tasks

- similarities on AXP and VAX (cont'd)
 - BACKUP command, 2-4
 - DECdtm, 2-5
 - decompressing libraries, 2-1
 - .EXE executable file type, 2-7
 - InfoServer supported on AXP, 2-4
 - LADCP supported on AXP, 2-4
 - LASTport network transport control program, 2-4
 - LAT startup, 2-4
 - LATCP, 2-4
 - LATSYM, 2-4
 - LTPAD process, 2-4
 - .OBJ object file type, 2-7
 - OPCOM, 2-5
 - RMS Journaling for OpenVMS, 2-3
 - STARTUP.COM, 2-1
 - SYCONFIG.COM, 2-1
 - SYLOGICAL.COM, 2-1
 - SYPAGSWPFILES.COM, 2-1
 - SYSECURITY.COM, 2-1
 - SYSTARTUP_V5.COM, 2-1
 - system directory structure, 2-1
 - two-phase commit protocol, 2-5
 - UETP (User Environment Test Package), 2-5
 - user-written device drivers, 2-3
 - VMScluster systems, 2-1
 - Volume Shadowing for OpenVMS, 2-3
- SHOW BUS command
 - See IO SHOW BUS command
- SHOW CPU command
 - /FULL qualifier on AXP, 2-18
- SHOW DEVICE command
 - See IO SHOW DEVICE command
- SHOW MEMORY command
 - displays AXP CPU-specific page values, 5-7
 - /GH_REGIONS qualifier, 5-15
- SHOW MEMORY/CACHE command
 - using to observe cache statistics, 5-17
- SHOW MEMORY/CACHE/FULL command
 - using to observe cache statistics, 5-17
- SHOW PREFIX command
 - See IO SHOW PREFIX command
- SHOW PROCESS command
 - /ALL qualifier
 - displays AXP pagelet values, 5-9
 - /CONTINUOUS qualifier
 - displays CPU-specific page values, 5-9
- SHOW QUEUE command
 - comparison on AXP and VAX, 3-6
 - /MANAGER qualifier, 3-6
- SHOW SECURITY command
 - /CLASS qualifier, 3-6
- SHOW SYSTEM command
 - /FULL on AXP displays CPU-specific page values and kilobyte values, 5-8
- SHOW WORKING_SET command
 - displays AXP pagelet and page values, 5-10
- SMP (symmetric multiprocessing)
 - comparison on AXP and VAX, 2-18
 - multiprocessing synchronization images, 2-18
 - on the DEC 4000 AXP, 2-18
 - on the DEC 7000 AXP, 2-18
- SMP_CPUS system parameter, 2-19
- SMP_LNGSPINWAIT system parameter, 2-19
- SMP_SANITY_CNT system parameter, 2-19
- SMP_SPINWAIT system parameter, 2-19
- SSNDJBC system service, 3-6
- SPTRREQ system parameter
 - obsolete on AXP, 5-6
- Starting network access, 6-2
- STARTNET.COM command procedure, 6-2
- START/QUEUE command
 - /AUTOSTART_ON qualifier, 3-5
 - comparison on AXP and VAX, 3-5
 - using AXP pagelet values, 5-11
- START/QUEUE/MANAGER command
 - comparison on AXP and VAX, 3-5
- Startup command procedure, 2-22
- STARTUP.COM command procedure, 2-1
- Streamlined multiprocessing synchronization images, 2-18
- SUBMIT command
 - /NOTE qualifier, 3-6
 - qualifiers using AXP pagelet values, 5-11
 - /RETAIN qualifier, 3-5
- SUMSLP utility (SUMSLP)
 - the same on VAX and AXP, 3-3
- SWPOUTPGCNT system parameter
 - changed default value on AXP, 5-6
 - dual values on AXP, 5-4
- SYCONFIG.COM command procedure, 2-1
- SYLOGICAL.COM command procedure, 2-1
- Symmetric multiprocessing
 - See SMP
- SYPAGSWPFILES.COM command procedure, 2-1
- SY\$BASE_IMAGE.EXE loadable executive image on AXP
 - new name for SYS.EXE, 2-26
- SYS.EXE loadable executive image on VAX
 - renamed to SY\$BASE_IMAGE.EXE on AXP, 2-26
- SYSECURITY.COM command procedure, 2-1
- SYSGEN (System Generation utility)
 - See System Generation utility
- SYSMAN (System Management utility)
 - See System Management utility

- SYSMWCNT system parameter
 - changed default value on AXP, 5-6
 - dual values on AXP, 5-3
- SYSPFC system parameter
 - dual values on AXP, 5-3
- SYSTARTUP_V5.COM command procedure, 2-1
- System directory structure, 2-1
- System Dump Analyzer (SDA) utility
 - CLUE commands, 3-9
- System Dump Analyzer utility (SDA)
 - conserving dump file storage space, 3-8
 - size of system dump files on AXP, 3-7
 - system dump file size requirement, 3-7
- System Generation utility (SYSGEN)
 - I/O subsystem configuration commands in SYSMAN utility on AXP, 2-15
 - parameters
 - See System parameters
 - system parameters in units of CPU-specific pages, 5-1
 - system parameters in units of pagelets, 5-1
 - system parameters on AXP, 2-15
- System management
 - differences on Alpha AXP and VAX, 1-2 to 1-6
- System management differences on AXP and VAX
 - changes to file names on AXP, 1-5
 - IO commands in SYSMAN, 1-4
 - layered product availability, 1-5
 - MONITOR POOL not necessary, 1-5
 - page size, 1-2
- System Management utility (SYSMAN)
 - comparison of I/O subsystem commands on AXP and VAX, 2-16
 - I/O configuration commands, A-1
 - I/O configuration support, A-1
 - I/O subsystem capabilities on AXP, 2-16
 - I/O subsystem configuration commands on AXP, 2-15
 - IO AUTOCONFIGURE command, A-1
 - IO CONNECT command, A-3
 - IO LOAD command, A-6
 - IO SET PREFIX command, A-6
 - IO SHOW BUS command, A-7
 - IO SHOW DEVICE command, A-8
 - IO SHOW PREFIX command, A-10
- System parameters
 - ACP_DINDXCACHE
 - unit change on AXP, 5-2
 - ACP_DIRCACHE
 - unit change on AXP, 5-2
 - ACP_HDRCACHE
 - unit change on AXP, 5-2
 - ACP_MAPCACHE
 - unit change on AXP, 5-2
 - ACP_WORKSET
 - unit change on AXP, 5-2
 - AWSMIN
 - System parameters
 - AWSMIN (cont'd)
 - changed default value on AXP, 5-6
 - dual values on AXP, 5-4
 - BALSETCNT
 - changed default value on AXP, 5-6
 - BORROWLIM
 - units that remain as CPU-specific pages on AXP, 5-3
 - changes in unit names only from VAX to AXP, 5-2
 - CLISYMTBL
 - changed default value on AXP, 5-6
 - unit change on AXP, 5-2
 - comparison of default values on AXP and VAX, 5-5
 - CPU-specific page units on AXP, 5-2
 - CTLIMGLIM
 - unit change on AXP, 5-2
 - CTLPAGES
 - unit change on AXP, 5-2
 - displaying
 - I/O subsystems, A-8
 - dual values on AXP, 5-3
 - DUMPSTYLE
 - changed default value on AXP, 5-7
 - controlling size of system dump files, 3-8
 - ERLBUFFERPAGES
 - unit change on AXP, 5-2
 - FREEGOAL
 - units that remain as CPU-specific pages on AXP, 5-3
 - FREELIM
 - units that remain as CPU-specific pages on AXP, 5-3
 - GBLPAGES
 - changed default value on AXP, 5-6
 - dual values on AXP, 5-3
 - GBLPAGFIL
 - changed default value on AXP, 5-6
 - units that remain as CPU-specific pages on AXP, 5-3
 - GH_EXEC_CODE, 5-14
 - GH_EXEC_DATA, 5-14
 - GH_RES_CODE, 5-14
 - GH_RES_DATA, 5-14
 - GH_RSRVPGCNT, 2-16, 5-14
 - GROWLIM
 - units that remain as CPU-specific pages on AXP, 5-3
 - IMGIOCNT
 - unit change on AXP, 5-2
 - LNMPHASHTBL
 - changed default value on AXP, 5-6
 - LNMSHASHTBL
 - changed default value on AXP, 5-6
 - MAXBUF
 - changed default value on AXP, 5-6

System parameters (cont'd)

measurement change for larger page size on
AXP, 5-1

MINWSCNT
unit change on AXP, 5-2

MPW_HILIMIT
changed default value on AXP, 5-6
units that remain as CPU-specific pages on
AXP, 5-3

MPW_LOLIMIT
changed default value on AXP, 5-6
units that remain as CPU-specific pages on
AXP, 5-3

MPW_LOWAITLIMIT
changed default value on AXP, 5-6
units that remain as CPU-specific pages on
AXP, 5-3

MPW_THRESH
changed default value on AXP, 5-6
units that remain as CPU-specific pages on
AXP, 5-3

MPW_WAITLIMIT
changed default value on AXP, 5-6
units that remain as CPU-specific pages on
AXP, 5-3

MPW_WRTCLUSTER
changed default value on AXP, 5-6
units that remain as CPU-specific pages on
AXP, 5-3

MULTIPROCESSING
on AXP and VAX, 2-15

NPAGEDYN
changed default value on AXP, 5-6

PAGEDYN
changed default value on AXP, 5-6

PAGTBLPFC
dual values on AXP, 5-3

PFCDEFAULT
dual values on AXP, 5-3

PHYSICAL_MEMORY
used on AXP instead of PHYSICALPAGES,
2-15

PIOPAGES
unit change on AXP, 5-2

POOLCHECK
adaptive pool management on AXP, 2-15

PQL_DBIOLM
changed default value on AXP, 5-6

PQL_DBYTLM
changed default value on AXP, 5-6

PQL_DDIOLM
changed default value on AXP, 5-7

PQL_DENQLM
changed default value on AXP, 5-7

PQL_DFILLM
changed default value on AXP, 5-7

PQL_DPGFLQUOTA
changed default value on AXP, 5-7

System parameters

PQL_DPGFLQUOTA (cont'd)
dual values on AXP, 5-4

PQL_DPRCLM
changed default value on AXP, 5-7

PQL_DTQELM
changed default value on AXP, 5-7

PQL_DWSDEFAULT
changed default value on AXP, 5-7
dual values on AXP, 5-4

PQL_DWSEXTENT
changed default value on AXP, 5-7
dual values on AXP, 5-4

PQL_DWSQUOTA
changed default value on AXP, 5-7
dual values on AXP, 5-4

PQL_MPGFLQUOTA
changed default value on AXP, 5-7
dual values on AXP, 5-4

PQL_MWSDEFAULT
changed default value on AXP, 5-7
dual values on AXP, 5-4

PQL_MWSEXTENT
changed default value on AXP, 5-7
dual values on AXP, 5-4

PQL_MWSQUOTA
changed default value on AXP, 5-7
dual values on AXP, 5-4

RSRVPAGCNT
units that remain as CPU-specific pages on
AXP, 5-3

SMP_CPUS, 2-19

SMP_LNGSPINWAIT, 2-19

SMP_SANITY_CNT, 2-19

SMP_SPINWAIT, 2-19

SPTREQ
obsolete on AXP, 5-6

SWPOUTPGCNT
changed default value on AXP, 5-6
dual values on AXP, 5-4

SYSMWCNT
changed default value on AXP, 5-6
dual values on AXP, 5-3

SYSPFC
dual values on AXP, 5-3

TBSKIPWSL
unit change on AXP, 5-2

VIRTUALPAGECNT
changed default value on AXP, 5-6
dual values on AXP, 5-4

WSDEC
dual values on AXP, 5-4

WSINC
dual values on AXP, 5-4

WSMAX
changed default value on AXP, 5-6
dual values on AXP, 5-4

T

Tailoring utilities

- See DECwindows
- See Tailoring utility

Tailoring utility (VMSTAILOR)
supported on AXP, 2-5

Tapes

- DSA local device naming, 2-23

Task-to-task communications, 6-2

TBSKIPWSL system parameter
unit change on AXP, 5-2

TECO editor, B-3

Terminal Fallback Facility (TFF), 2-27

Terminal Fallback Utility (TFU), 2-27

TFF (Terminal Fallback Facility)

- See Terminal Fallback Facility

TFU (Terminal Fallback Utility)

- See Terminal Fallback Utility

Tuning

- See Performance optimization tasks

Tuning system parameters

- on AXP, 5-1 to 5-11

Two-phase commit protocol

- DECdtm supported on AXP, 2-5

U

UETPs (User Environment Test Packages)

- similar on VAX and AXP, 2-5

Uniprocessing synchronization images, 2-18

UNLOCK command

- not supported on AXP, B-2

Upline dumping, 6-2

User Environment Test Packages

- See UETPs

User-written device drivers, 2-3

USER_MSGS boot flag, 2-11

V

VAX C

- See C programming language

VAX systems

- differences in system management from AXP
overview, 1-2
- page size, 1-3

VCC_FLAGS system parameter, 5-17

VCC_MAXSIZE system parameter, 5-17

Vectors

- MONITOR VECTOR, 3-2
- not in AXP computers, 3-2

Virtual I/O caches, 5-17

VIRTUALPAGECNT system parameter

- changed default value on AXP, 5-6
- dual values on AXP, 5-4

VMScLuster environments

- cluster aliases supported on AXP, 6-1
- features, 2-2
- PAK name differences, 2-14
- restriction for using multiple queue managers,
3-7
- similarities on AXP and VAX, 2-1
- symmetric multiprocessing supported on AXP,
2-18

VMSINSTAL procedure

- comparison with POLYCENTER Software
Installation utility, 2-7

VMSINSTAL.COM command procedure

- history file of VMSINSTAL executions, 2-24
- INSTALLED_PRDS.COM command procedure,
2-25

listing installed products, 2-24

product installation log file, 2-24, 2-25

VMSINSTAL.HISTORY history file, 2-25

VMSINSTAL.HISTORY history file, 2-25

VMSTAILOR

- See Tailoring utility

Volume Shadowing

- similarities on AXP and VAX, 2-3

W

WSDEC system parameter

- dual values on AXP, 5-4

WSINC system parameter

- dual values on AXP, 5-4

WSMAX system parameter

- changed default value on AXP, 5-6
- dual values on AXP, 5-4

X

X.25 routers

- clearing DTE counters, 6-8
- connections not supported on AXP, 6-5
- protocol module counters, 6-8
- protocol module parameters, 6-6
- server module
 - clearing call handler counters, 6-8
 - counters, 6-8
- server module parameters, 6-6
- testing lines
 - not supported on AXP, 6-6

Z

ZERO MODULE X25-PROTOCOL command, 6–8

ZERO MODULE X25-SERVER command, 6–8

ZERO MODULE X29-SERVER command, 6–8

