DEC Ada

Installation Guide for OpenVMS VAX Systems

Order Number: AA–EF85G–TE

December 1998

This guide contains instructions for installing DEC Ada Version 3.5 or higher on OpenVMS VAX systems. It also explains how to read the online release notes before or after installing the product.

Revision/Update Information:	This revised guide supersedes the *DEC Ada Installation Guide for OpenVMS VAX Systems* (Order No.: AA–EF85F–TE).
Operating System & Version:	OpenVMS VAX Version 6.2 through Version 7.2
Software Version:	DEC Ada Version 3.5

Digital Equipment Corporation
Maynard, Massachusetts

December 1998

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 authorizied only pursuant to a valid written license from DIGITAL or an authorized sublicensor.

© Digital Equipment Corporation 1998. All Rights Reserved.

The following are trademarks of Digital Equipment Corporation: Bookreader, DEC, DEC Ada, DEC Ada PDO, DECnet, DECwindows, DIGITAL, OpenVMS, VAX, VAX DOCUMENT, OpenVMS Cluster, XD Ada, and the DIGITAL logo.

The following are third-party trademarks:

IEEE is a registered trademark of the Institute of Electronic and Electrical Engineers.

Intel is a registered trademark of Intel Corporation.

Motif, OSF, and OSF/1 are registered trademarks of the Open Software Foundation, Inc.

POSIX is a registered trademark of IEEE.

POSTSCRIPT is a registered trademark of Adobe Systems, Inc.

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

VADS is a registered trademark of Rational Software Corporation.

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

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

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

This document is also available on CD–ROM.

This document was prepared using VAX DOCUMENT Version 2.1.

Contents

Preface vii

1 Preparing for DEC Ada Installation

1.1 Required Operating System and Prerequisite Hardware and Software 1–1
1.2 License Registration 1–2
1.3 OpenVMS Cluster Considerations 1–2
1.4 Installation Procedure Requirements 1–3
1.4.1 Privileges and Disk Space 1–3
1.4.2 System Parameters 1–3
1.4.2.1 Checking UAF Parameters 1–4
1.4.2.2 Calculating the Values for GBLPAGES and GBLSECTIONS 1–4
1.4.2.3 Changing System Parameter Values with AUTOGEN . . . 1–5
1.4.3 Process Account Quotas 1–6
1.4.4 VMSINSTAL Requirements 1–7
1.4.5 Backing Up Your System Disk 1–7

2 Installing DEC Ada

2.1 The Installation Procedure 2–1
2.1.1 Invoke VMSINSTAL 2–1
2.1.2 Confirm System Backup 2–3
2.1.3 Mount the Media 2–3
2.1.4 Select a Release Notes Option 2–4
2.1.5 Choose the Installation Verification Procedure Option 2–5
2.1.6 Select Installation Options 2–5
2.1.7 Read Informational Messages 2–6
2.1.8 Observe the Installation Verification Procedure 2–7
2.1.9 End the Installation Procedure 2–7
2.2 Error Recovery 2–8
2.3 Keeping the Currently Existing Compiler Available 2–8

2.3.1 Saving the Currently Existing Help File 2–11
2.3.2 Saving the Currently Existing Examples Directory 2–12
2.4 Installing DEC Ada on an OpenVMS Cluster 2–12
2.5 Postinstallation Notes 2–12

3 After Installing DEC Ada

3.1 Running the Installation Verification Procedure Separately 3–1
3.2 Error Conditions 3–1
3.3 Installing Updates 3–2
3.4 Determining and Reporting Problems 3–2

A Sample Installation

B Files and Logical Names Installed

B.1 File Names B–1
B.2 Logical Names B–2

C Tuning OpenVMS VAX Systems

C.1 General Tuning of OpenVMS Systems C–1
C.2 Improving Product Performance C–1
C.2.1 ASTLM C–2
C.2.2 ENQLM C–3
C.2.3 FILLM C–3
C.2.4 PRCLM C–4
C.2.5 TQELM C–4
C.2.6 Virtual Memory Usage C–4
C.2.7 Batch Queue Parameters C–6
C.2.8 WSMAX Parameter C–7
C.2.9 Channel Count Parameters C–7
C.2.10 Program Library Networking Effects C–7
C.3 Relevant File Quotas and Cache Sizes C–8
C.3.1 Caching Files C–9
C.3.2 Caching File Headers C–9
C.3.3 Caching Directory Files C–11
C.4 Using DECram on OpenVMS Systems C–11

Figures

1	Documentation Reading Path for Related Documents	viii
2	Documentation Reading Path for DEC Ada Documentation	ix

Tables

1–1	Disk Space Requirements at Installation	1–3
1–2	Minimum Required System Parameter Values	1–4
1–3	Process Account Quotas for the Installing Account	1–6
B–1	DEC Ada Logical Names and Locations	B–2

Preface

This guide explains how to install DEC Ada Version 3.5 or higher on OpenVMS VAX systems.

Keep this guide with your distribution kit. You need it to install maintenance updates or to reinstall DEC Ada for any other reason.

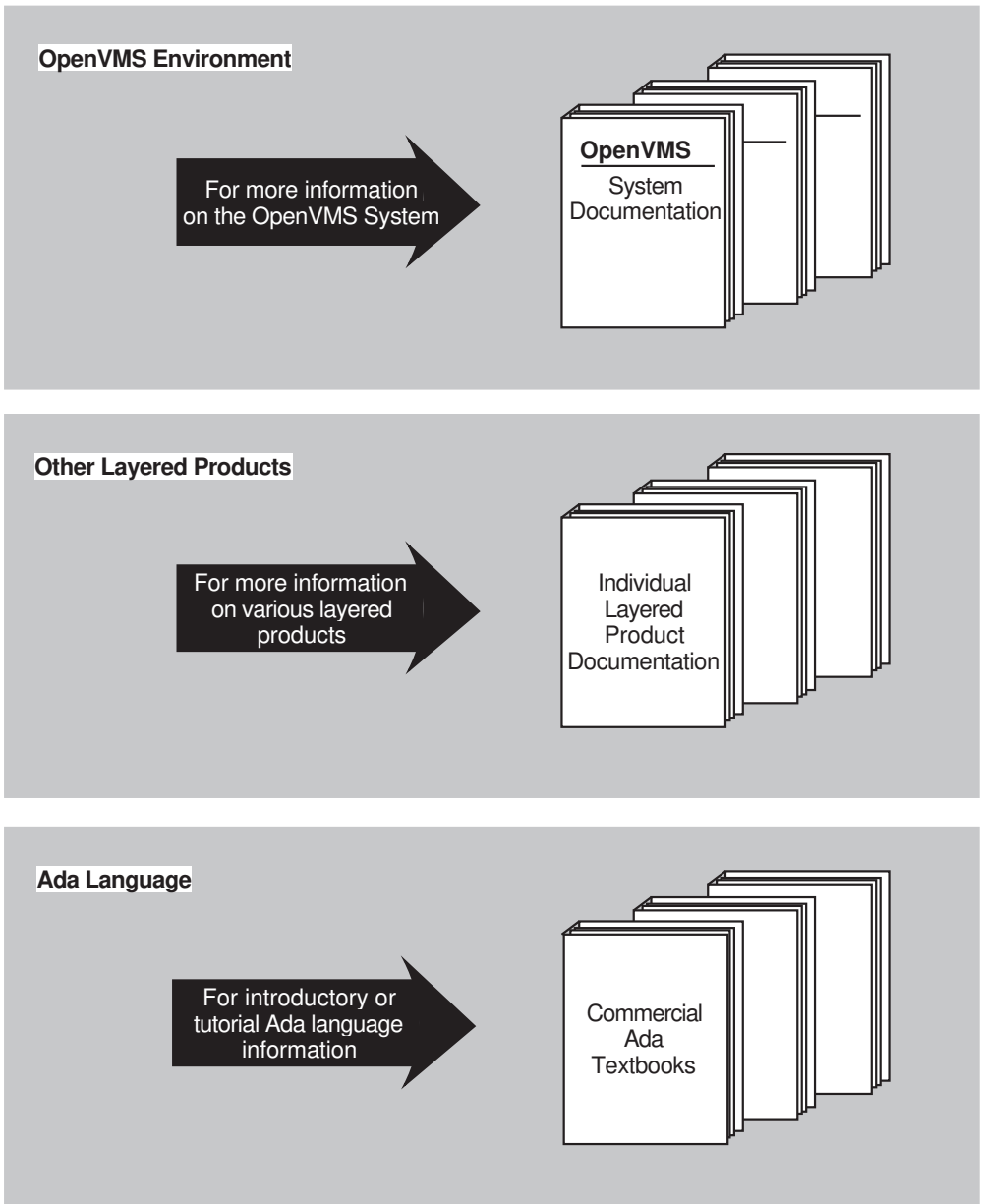
Intended Audience

This guide is intended for the system manager who installs DEC Ada software.

Documentation Reading Path

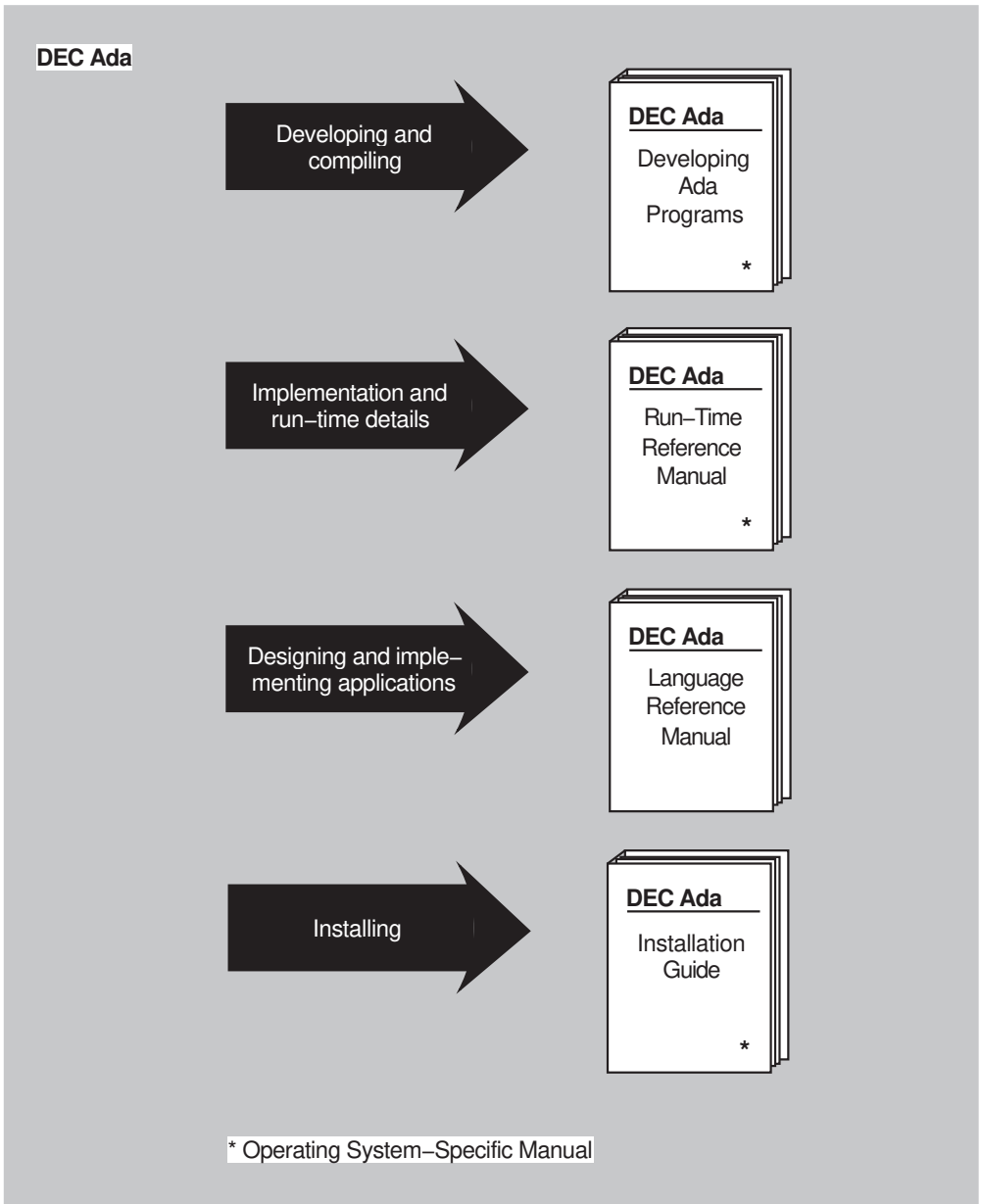
Figures 1 and 2 show the relationship of the Ada documentation set to other documentation that may be helpful.

Figure 1 Documentation Reading Path for Related Documents



ZK–5349A–2–GE

Figure 2 Documentation Reading Path for DEC Ada Documentation



ZK–5349A–1–GE

Document Structure

This guide contains the following chapters and appendixes:

- Chapter 1 describes the operating system and hardware requirements for DEC Ada installation and related procedures that you complete before installing DEC Ada.
- Chapter 2 describes the step-by-step instructions for the installation.
- Chapter 3 describes postinstallation actions and considerations.
- Appendix A contains the output from a sample DEC Ada installation.
- Appendix B contains a list of files installed and logicals defined as a result of the installation procedure.
- Appendix C describes how to tune your system after you have installed DEC Ada.

Getting Help from DIGITAL

If you have a customer support contract and have comments or questions about DEC Ada software, you can contact the DIGITAL Customer Support Center (CSC), preferably using electronic means such as DSNlink. In the United States, customers can call the CSC at 1–800–354–9000.

In addition, DIGITAL now provides access to Web Information and Support (WIS), a new electronic delivery application that allows you to submit electronic service requests, to access databases, to download patch kits, to transfer files from DIGITAL, and to connect to global notes conferences via the World Wide Web. If you are an entitled service customer with access to the Web, you can now use your favorite browser to connect to electronic services through an easy-to-use, graphical interface.

If you are already a DIA user, you can gain access to WIS by opening http://www.service.digital.com/wis. Click on the words ‘‘United States,’’ and enter your DIA Access ID and password. WIS and DIA use the same underlying account and directory on the DIGITAL Customer Support Center host system.

If you are not already registered for DIA, you can register for a WIS account by contacting your Customer Support Center or local DIGITAL representative.

Conventions

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

References to OpenVMS refer to both OpenVMS Alpha and OpenVMS VAX. Specific references to either platform are noted.

The name of the DEC OSF/1 operating system has been changed to the DIGITAL UNIX operating system. Any references to DEC OSF/1 are synonymous with DIGITAL UNIX unless specified otherwise.

References to the ULTRIX operating system are to ULTRIX MIPS systems, not ULTRIX VAX systems.

The following conventions are also used in this guide:

Convention	Description
$	A dollar sign ($) represents the OpenVMS DCL system prompt.
Ctrl/*x*	A sequence such as Ctrl/*x* indicates that you must hold down the key labeled Ctrl while you simultaneously press another key or a pointing device button.
Return	In examples, a key name enclosed in brackets or a box indicates that you press a key on the keyboard.
boldface monospace text	In interactive examples, boldface monospace text represents user input.
`Monospace type`	Monospace type indicates code examples and interactive screen displays. This typeface is also used in text to indicate the exact name of a command, routine, partition, path name, directory, or file.
file-spec . . .	A horizontal ellipsis following a parameter, option, or value in syntax descriptions indicates that additional parameters, options, or values can be entered.
. . .	A horizontal ellipsis in an Ada example or figure indicates that not all of the statements are shown.
. . .	A vertical ellipsis in an interactive figure or example indicates that not all of the commands and responses are shown.

Convention	Description
italic text	Italic text emphasizes important information, indicates variables, and refers to complete titles of manuals. Italic text also represents information that can vary in system messages (for example, Internal error *number*.)
n	A lowercase italic *n* indicates the generic use of a number.
Variable	This typeface is used in syntax to indicate variable values.
UPPERCASE TEXT	Uppercase indicates the name of a command, routine, parameter, procedure, utility, file, file protection code, or the abbreviation for a system privilege.
boldface text	Boldface text indicates DEC Ada reserved words.
*type*_name	Italicized words in syntax descriptions indicate descriptive prefixes that are intended to give additional semantic information rather than to define a separate syntactic category.
[expression]	Square brackets indicate that the enclosed item is optional.
{, mechanism_name }	Braces in DEC Ada syntax indicate that the enclosed item can be repeated zero or more times.
|	A vertical bar in DEC Ada syntax separates alternative items unless it occurs immediately after an opening brace, in which case it stands for itself.

1
Preparing for DEC Ada Installation

This chapter discusses the preparations and requirements necessary for installing DEC Ada on OpenVMS VAX systems.

The information in the installation guide pertains to DEC Ada Version 3.5 or higher for OpenVMS VAX systems. DIGITAL suggests that you upgrade your DEC Ada compiler and compilation environment incrementally with each new release, but there is no technical requirement to do so. It is required that you review carefully the release notes for each version because they capture important release-oriented information and advise you of changes between versions.

Your bill of materials (BOM) and indented bills report (BIL) specify the number and contents of your media. Be sure to verify the contents of your kit with this information. If your kit is damaged or if you find that parts of it are missing, contact your DIGITAL representative.

Note

DEC Ada provides online release notes. They are contained in the kit with the compiler and its associated components. DIGITAL strongly recommends that you read the release notes before proceeding with the installation. For information on accessing the online release notes, see Section 2.1.

1.1 Required Operating System and Prerequisite Hardware and Software

DEC Ada Version 3.5 supports OpenVMS VAX Version 6.2 through Version 7.2.

For a complete list of the required operating system components, current hardware requirements, and prerequisite and optional software and their required version numbers, see the Software Product Description (SPD), which is shipped on the DEC Ada kit in SYS$HELP:ADA03*n*_SPD.*.

1.2 License Registration

If you are installing DEC Ada on a newly licensed node or cluster, you must first register a License Product Authorization Key (License PAK) using the License Management Facility (LMF). The License PAK can be shipped along with the kit if you ordered the license and media together. Otherwise, the PAK is shipped separately to a location based on your license order.

If you are installing an update of DEC Ada but lack a service contract, call your DIGITAL representative for instructions on how to get a License PAK.

If you are installing prerequisite or optional software along with DEC Ada, review the PAK status and install the PAKs for any prerequisite or optional software before you install DEC Ada.

You must register and load your license for DEC Ada *before* you start the installation. To register a license under OpenVMS, first log in to the system manager's account, SYSTEM. You then have a choice of two ways to perform the registration:

- Invoke the SYS$UPDATE:VMSLICENSE.COM procedure. When it prompts you for information, respond with data from your License PAK.
- At the DCL prompt ($), enter the LICENSE REGISTER command with the appropriate qualifiers that correspond to License PAK information.

If you plan to use DEC Ada on more than one node in an OpenVMS Cluster, you need to perform a license load on the other nodes after you complete this installation.

For complete information on using LMF, see the *OpenVMS License Management Utility Manual*.

1.3 OpenVMS Cluster Considerations

If you are running DEC Ada on an OpenVMS Cluster, VMSINSTAL automatically installs new versions as known or shared images only on the machine on which the installation is actually performed. You must manually install new versions as known or shared images on other machines using the OpenVMS Install utility or invoke SYS$STARTUP:ADA$STARTUP.COM on each of the other cluster members.

1.4 Installation Procedure Requirements

This section and those that follow discuss various requirements for installing DEC Ada on OpenVMS VAX systems.

1.4.1 Privileges and Disk Space

To install DEC Ada, you must be logged in to an account that has SETPRV or at least the following privileges:

- CMKRNL
- WORLD
- SYSPRV

VMSINSTAL turns off BYPASS privilege at the start of the installation.

The installation checks the system disk for the space it requires to install DEC Ada. For a summary of the storage requirements, refer to the SPD.

To determine the number of free disk blocks on the current system disk, enter the following command at the DCL prompt ($):

```
$ SHOW DEVICE SYS$SYSDEVICE
```

Table 1–1 identifies the various installation scenarios and the corresponding disk space requirements.

Table 1–1 Disk Space Requirements at Installation

Scenario	Blocks Required During Installation	Blocks Required After Installation
DEC Ada never previously installed	105,000	76,000
DEC Ada Version 3.4 previously installed, *not* saving previous product	69,000	15,000

1.4.2 System Parameters

Installing DEC Ada requires certain system parameter settings. Table 1–2 lists the minimum required system parameter values for the installation. Depending on the type of programs and applications running at your site, you might need higher values for some settings.

Table 1–2 Minimum Required System Parameter Values

System Parameter	Minimum Value
GBLPAGES[1]	11000 + DCLTABLES
GBLSECTIONS[1]	10

[1]The values listed for these system parameters represent the number of free global pages and global sections required for the installation, not the total number you need to run your system and other software.

To use DEC Ada, each account or batch queue that invokes the DEC Ada compiler must have TMPMBX and NETMBX privileges.

The following sections show you how to:

- Check UAF parameters
- Calculate values for the GBLPAGES and GBLSECTIONS system parameters
- Change parameter values with the AUTOGEN command procedure

1.4.2.1 Checking UAF Parameters

Before you install DEC Ada, you should check whether your account has sufficient quotas to perform the installation. You can check the quotas by entering the following command at the DCL prompt to invoke the AUTHORIZE utility:

```
$ RUN SYS$SYSTEM:AUTHORIZE
UAF> SHOW SYSTEM
```

After checking your quotas, enter the EXIT command at the UAF> prompt to return to DCL level.

1.4.2.2 Calculating the Values for GBLPAGES and GBLSECTIONS

To install and run DEC Ada, you must have sufficient free global pages and global sections. You must first find out how many free global pages and sections you have on your system. Then use AUTOGEN if you need to increase the GLBPAGES and GBLSECTIONS system parameters.

For example, you can use the WRITE command with the F$GETSYI lexical function to find the number of free global pages and global sections. The following example shows how to get this information at your terminal (the default for SYS$OUTPUT):

```
$ WRITE SYS$OUTPUT F$GETSYI("FREE_GBLPAGES")
15848
$ WRITE SYS$OUTPUT F$GETSYI("FREE_GBLSECTS")
24
```

If the values the system displays are greater than the values in Table 1–2, you do not need to change the settings. If the value of free global pages or global sections is less than the value in Table 1–2, you must increase the system parameter setting.

Section 1.4.2.3 describes the procedures for increasing these values using AUTOGEN.

1.4.2.3 Changing System Parameter Values with AUTOGEN

Use the AUTOGEN command procedure to change system parameters. AUTOGEN automatically adjusts values for parameters that are associated with the values you reset manually. To change system parameters with AUTOGEN, edit the following file:

```
SYS$SYSTEM:MODPARAMS.DAT
```

To change a parameter value listed in this file, delete the current value associated with that parameter and enter the new value.

To add a new parameter, add a line to the file that includes both the name of the parameter and its value. For example:

```
WSMAX = 1024
```

To modify incremental parameters such as GBLPAGES and GBLSECTS, use ADD_. The following example increases the global page setting by 2000:

```
ADD_GBLPAGES = 2000
```

When you set the page file quota, be careful not to use a value that exceeds the amount of page file space available on the system.

After you make all your changes, exit from the editor and execute the AUTOGEN procedure to recalculate your system parameters. Enter the following command at the DCL prompt:

```
$ @SYS$UPDATE:AUTOGEN GETDATA REBOOT
```

When you specify REBOOT, AUTOGEN performs an automatic system shutdown and then reboots the system. Any users logged on to the system are immediately disconnected during the shutdown. The automatic reboot puts the new parameter values into effect.

The AUTOGEN command procedure automatically adjusts some of the SYSGEN parameters based on the consumption of resources since the last reboot. If you do not want to take advantage of this automatic adjustment, include the NOFEEDBACK qualifier on the AUTOGEN command line.

For more information about using AUTOGEN, see the system management documentation for the OpenVMS operating system.

1.4.3 Process Account Quotas

The account you use to install DEC Ada must have sufficient quotas to enable you to perform the installation. Table 1–3 summarizes the minimum process quotas required for the installation account. Larger quotas may be required for other work done by this account. For more information about these process quotas and the relationships among them, see Appendix C.

Table 1–3 Process Account Quotas for the Installing Account

Account Quota	Value
ASTLM	24
BIOLM	18
BYTLM	18000
DIOLM	18
ENQLM	300
FILLM	100

User account quotas are stored in the file SYSUAF.DAT. Use the Authorize utility (AUTHORIZE) to verify and change user account quotas. First, set your directory to SYS$SYSTEM and then, run AUTHORIZE:

```
$ SET DEFAULT SYS$SYSTEM
$ RUN AUTHORIZE
UAF>
```

At the UAF> prompt, enter the SHOW command with an account name to check a particular account. For example:

```
UAF> SHOW SMITH
```

To change a quota, enter the MODIFY command. The following example changes the FILLM quota for the SMITH account and then exits from the utility:

```
UAF> MODIFY SMITH /FILLM=50
UAF> EXIT
```

After you exit from the utility, the system displays messages indicating whether or not changes were made. Once the changes have been made, you must log out and log in again for the new quotas to take effect.

For more information on modifying account quotas, see the description of the Authorize utility in the system management documentation for the OpenVMS operating system.

1.4.4 VMSINSTAL Requirements

When you invoke VMSINSTAL, it checks the following:

- Whether you are logged in to a privileged account
- Whether any users are logged in to the system
- Whether you have adequate quotas for installation

If problems are detected during installation, in some instances you can enter Yes to continue. If you do not wish to continue, enter No or press Return to stop the installation process. Then correct the problem and restart the installation.

1.4.5 Backing Up Your System Disk

At the beginning of the installation, VMSINSTAL asks if you have backed up your system disk. DIGITAL recommends that you do a system disk backup before installing any software.

Use the backup procedures established at your site. For details on performing a system disk backup, see the section on the OpenVMS Backup utility in the system management documentation for the OpenVMS operating system.

2
Installing DEC Ada

This chapter explains how to install DEC Ada on OpenVMS VAX systems. Section 2.1 contains a step-by-step description of the installation procedure.

2.1 The Installation Procedure

DEC Ada Version 3.5 (ADA035) contains three savesets (A, B, and C). The DEC Ada installation procedure consists of a series of questions and informational messages. See Appendix A for the output from a sample installation.

The procedure takes approximately 15 minutes to complete.

To abort the installation procedure at any time, press Ctrl/Y. When you press Ctrl/Y, the installation procedure deletes all files that it has created up to that point and exits. You can then start the installation again.

When the system prompts you with a question during the installation procedure, the default answer is often listed in brackets ([ ]).

2.1.1 Invoke VMSINSTAL

To start the installation, invoke the VMSINSTAL command procedure from a privileged account, such as the SYSTEM account. The command format is as follows:

@SYS$UPDATE:VMSINSTAL saveset-name device:[pathname] OPTIONS N

VMSINSTAL is in the SYS$UPDATE directory. To install DEC Ada, enter the following command:

```
$ @SYS$UPDATE:VMSINSTAL ADA03n DKA100: OPTIONS N
```

The following defines the elements of the VMSINSTAL command procedure:

saveset-name
The installation name for the component. For DEC Ada, use the installation name ADA03*n*.

n
The letter *n* represents the maintenance update of DEC Ada. See the label on the DEC Ada distribution kit or the bill of materials for the maintenance update number of your kit. For DEC Ada Version 3.5, the maintenance update is 5.

device:[pathname]
The name of the device on which you plan to mount the media. For example, DKA100: is the device name for a disk drive. It is not necessary to use the console drive for this installation. However, if you do use the console drive, you should replace any media you removed once the installation is complete.

OPTIONS N
An optional parameter that indicates that you want to be asked to choose one of the four options for reviewing the release notes. (See Section 2.1.4). If you do not include the OPTIONS N parameter, VMSINSTAL does not ask you about the release notes.

You should review the release notes before proceeding with the installation in case they contain additional information about the installation. If you are restarting the installation and have already reviewed the release notes, you do not need to specify OPTIONS N.

You can select several other options when you invoke the VMSINSTAL command procedure:

- Option (A): Auto_answer

  Initially, creates a file that contains your answers to VMSINSTAL questions and prompts. You can then use the option (and the answer file) to save time during a reinstallation (typically after upgrading your system).

- Option (G): Get save set

  Lets you store product save sets temporarily on a magnetic tape or in a disk directory.

- Option (L): File log

  Logs all activity to the terminal during installation.

See the material on software installation in the system management documentation for the OpenVMS operating system for detailed information on these options. If you specify more than one option, separate each option name with a comma (for example, OPTIONS A,N).

The following example invokes VMSINSTAL to install DEC Ada from disk drive DKA100: and shows the system response. This example uses the OPTIONS N release note parameter.

```
$ @SYS$UPDATE:VMSINSTAL ADA03n DKA100: OPTIONS N

      OpenVMS VAX Software Product Installation Procedure Vm.n


It is dd-mmm-yyyy at hh:mm.
Enter a question mark (?) at any time for help.
```

If you do not supply either the product name or the device name, VMSINSTAL prompts you for options.

2.1.2 Confirm System Backup

VMSINSTAL asks if you are satisfied with your system backup.

```
* Are you satisfied with the backup of your system disk [YES]?
```

You should always back up your system disk before performing an installation. If you are satisfied with the backup of your system disk, press Return. Otherwise, enter No to discontinue the installation. After you back up your system disk, you can restart the installation.

2.1.3 Mount the Media

You should now mount the first distribution volume on the device you specified when you invoked VMSINSTAL. The device name appears in the line preceding the question. VMSINSTAL then asks you if you are ready to continue with the installation.

If you respond Yes to indicate that you are ready, VMSINSTAL displays a message that the media containing DEC Ada has been mounted on the specified device and that the installation has begun.

If you entered the wrong device name when you invoked VMSINSTAL and need to restart the installation, enter No in response to the ‘‘Are you ready?’’ question. To abort the installation for other reasons, press Ctrl/Y.

```
Please mount the first volume of the set on DKA100:.
* Are you ready? YES
%MOUNT-I-MOUNTED, DEC Ada MOUNTED ON _$$DKA100:  (NODE 1)
The following products will be processed:
  DEC Ada Version 3.5

    Beginning installation of DEC Ada Version 3.n at 20:01
```

2.1.4 Select a Release Notes Option

If you specified OPTIONS N when you invoked VMSINSTAL, you are now asked to choose one of the four options for reviewing the release notes.

```
Release notes included with this kit are always copied to SYS$HELP.

Additional Release Notes Options:

 1.  Display release notes
 2.  Print release notes
 3.  Both 1 and 2
 4.  None of the above

* Select option [2]:
```

If you select option 1, VMSINSTAL displays the release notes immediately on the console terminal. You can terminate the display at any time by pressing Ctrl/C.

If you select option 2, VMSINSTAL prompts you for the name of the print queue that you want to use to print the release notes:

```
* Queue name [SYS$PRINT]:
```

You can press Return to send the file to the default output print device or you can enter another queue name.

If you select option 3, VMSINSTAL displays the release notes immediately on the console terminal and then prompts you for a queue name for the printed version.

Select option 4 if you have already reviewed the release notes and are continuing the installation.

Next VMSINSTAL displays the following question:

```
* Do you want to continue the installation [N]?: YES
%VMSINSTAL-I-RELMOVED, The product's release notes have been
                                 moved to SYS$HELP.
```

To continue the installation, enter Yes. Otherwise, press Return if you want to stop the installation to read the release notes. In either case, the release notes

are copied to a file in the SYS$HELP directory:

```
SYS$HELP:ADA03n.RELEASE_NOTES
```

The release notes are also available in POSTSCRIPT format. The file name is SYS$HELP:ADA03*n*.RELEASE_NOTES_PS.

2.1.5 Choose the Installation Verification Procedure Option

The installation procedure now asks if you want to run the Installation Verification Procedure (IVP).

```
* Do you want to run the IVP after the installation [YES]?
```

The IVP for DEC Ada checks to be sure that the installation is successful. It is recommended that you run the IVP.

After installing DEC Ada, you can run the IVP independently to verify that the software is available on your system. You might need to run the IVP after a system failure to make sure that users can access DEC Ada.

For more information on IVP, see Section 3.1.

2.1.6 Select Installation Options

DEC Ada provides support for the DEC Source Code Analyzer (SCA). SCA is an optional, interactive, multilanguage, source code cross-reference and static analysis tool. The information in the SCA library for the predefined units allows you to query SCA about symbols, declarations, and subprograms contained in the predefined units.

If you have SCA installed on your system and intend to use it to analyze DEC Ada programs, type Yes in response to the prompt. If you are not going to use SCA or you are concerned about saving disk space, type No in response to the prompt.

```
* Do you want to include SCA analysis data file for the predefined
  units [YES]? Return
```

If you need to know how much disk space this library will take before you can answer the prompt, type a question mark (?) to obtain specific information on disk space requirements before typing Yes or No.

If you typed Yes or pressed Return, an SCA analysis data file for the DEC Ada predefined units (ADA$SCA_PREDEFINED.ANA) is copied to the SYS$LIBRARY directory during installation. To make this information available, you must create an SCA library and load the data from the file into the library. Previously, a complete SCA library for the predefined units was installed.

If the current version of DEC Ada on your system is different from the version you are installing, the installation procedure allows you to save the currently existing version. If you want to save the currently existing compiler, answer Yes. If not, answer No to the prompt.

```
* Do you want to save the currently existing compiler [YES]?
```

For more information about keeping the currently existing compiler available, see Section 2.3.

If you need to know exactly which files are saved, then type a question mark (?) to obtain specific information before typing Yes or No.

2.1.7 Read Informational Messages

At this point, the installation procedure displays a number of informational messages that report on the progress of the installation. There are no further questions.

If the installation procedure has been successful up to this point, VMSINSTAL moves the new or modified files to their target directories, updates help files, and updates DCL tables if necessary.

If you chose to have files purged, that work is done now. The following messages are displayed:

```
    There will not be any further questions.

%VMSINSTAL-I-RESTORE, Restoring product save set B ...

    After the installation is complete, you need to add the following
    command to the site-specific system startup command file,
    SYS$MANAGER:SYSTARTUP_VMS.COM:
 $ @SYS$STARTUP:ADA$STARTUP


%VMSINSTAL-I-SYSDIR, This product creates system directory
                                        [SYSHLP.EXAMPLES.ADA].
%VMSINSTAL-I-RESTORE, Restoring product save set C ...
%ACS-I-CL_VFYOK, VMI$ROOT:[SYSLIB.ADALIB$NEW] verified
```

The success of the installation is indicated by the following message:

```
%VMSINSTAL-I-MOVEFILES, Files will now be moved to their target
                                                  directories...
```

If the installation procedure is successful, the procedure moves the files created to their target directories.

At this point, an informational message notifies you when the new files are moved to their target directories. If you are installing DEC Ada on a system that has close to the minimum amount of free disk space, the created files are moved to their target directories as they are processed and no message is issued when the files are moved.

For information on the files that the installation procedure creates or modifies, see Appendix B.

2.1.8 Observe the Installation Verification Procedure

If you chose to run the IVP, VMSINSTAL runs it at this point. When the IVP runs successfully, you see the following display:

```
%ACS-I-CL_LIBCRE, Library XDFRZ$DKB100:[SYS0.SYSUPD.ADA03n.ADALIB] created
%ACS-I-CL_LIBIS, Current program library is
        XDFRZ$DKB100:[SYS0.SYSUPD.ADA03n.ADALIB]
DEC Ada V3.n installation verification test
Testing, testing,           1,           2,           3,           4
Testing, testing
DEC Ada V3.n installation verification PASSED
%ACS-I-CL_LIBDEL, Library XDFRZ$DKB100:[SYS0.SYSUPD.ADA03n.ADALIB] deleted
        Installation of ADA V3.n completed at 11:18
```

After the IVP has completed successfully, the temporary program library and any files generated during the IVP are deleted.

2.1.9 End the Installation Procedure

The following messages indicate that the entire installation procedure is complete.

```
   Installation of DEC Ada 3.n completed at hh:mm

   VMSINSTAL procedure done at hh:mm
```

You can now log out of the privileged account:

```
$ LOGOUT Return
SYSTEM     logged out at dd-mmm-yyyy hh:mm:ss.ss
```

When the product installation procedure is complete, you can choose to install more products or to log out (as shown here.) If you removed any media from the console drive before beginning the installation, you should replace it now.

VMSINSTAL deletes or changes entries in the process symbol tables during the installation. If you are going to continue using the system manager's account and you want to restore these symbols, you should log out and log in again.

2.2 Error Recovery

If errors occur during the installation itself or when the Installation Verification Procedure (IVP) is running, VMSINSTAL displays failure messages. If the installation fails, you see the following message:

```
%VMSINSTAL-E-INSFAIL, The installation of DEC Ada 3.n has failed.
```

If the IVP fails, you see the following messages:

```
The DEC Ada 3.n Installation Verification Procedure failed.

%VMSINSTAL-E-IVPFAIL, The IVP for DEC Ada 3.n has failed.
```

Errors can occur during the installation if any of the following conditions exist:

- The operating system version is incorrect.
- A prerequisite software version is incorrect.
- Quotas necessary for successful installation are insufficient.
- System parameter values for successful installation are insufficient.
- The OpenVMS Help library is currently in use.
- The product license has not been registered and loaded.

For descriptions of the error messages generated by these conditions, see the OpenVMS documentation on system messages, recovery procedures, and OpenVMS software installation. If you are notified that any of these conditions exist, you should take the appropriate action as described in the message. For information on installation requirements, see Chapter 1.

2.3 Keeping the Currently Existing Compiler Available

The installation procedure installs the compiler and program library manager on the kit as the default compiler and program library manager. When you enter an ADA or ACS command, the newly installed compiler and program library manager are invoked. The DEC Ada kit is designed to let you have continued access to your previously existing compiler while you are using the newly installed compiler.

During the installation, you are asked if you want to save the currently existing compiler. If you answer Yes and the version of the currently existing compiler is not the same as the version of the compiler that you are installing, the installation procedure creates a subdirectory of SYS$SYSTEM. The subdirectory is named ADA$*ident_string*.DIR, where *ident_string* is the identifying string of the currently existing compiler. The installation procedure

then copies the currently existing versions of the following files to that subdirectory:

- Compiler (ADA.EXE)
- Program library manager (ACS.EXE)
- Command definition file (ADACLD.CLD)
- Run-time library (ADARTL.EXE)
- Message file (ADAMSG.EXE)
- Library of predefined units (ADALIB.DIR)

In addition, the installation procedure places the following command procedures in SYS$SYSTEM:

- ADA$USE_*ident_string*.COM
- ADA$USE_INSTALLED.COM

When invoked, ADA$USE_*ident_string*.COM defines the appropriate logical names needed to access the previously existing compiler. ADA$USE_ INSTALLED.COM defines the appropriate logical names to access the installed compiler.

For example, if you install Version 3.5 as the default compiler and you want to access the Version 3.4 compiler, you enter the following command:

```
$ @SYS$SYSTEM:ADA$USE_V3_4.COM
```

Subsequently, when you enter an ADA or ACS command, the Version 3.4 compiler or program library manager is invoked.

If you want the old version of the DEC Ada compiler to be the default, modify SYS$STARTUP:ADA$STARTUP.COM by replacing the current logical definitions with the ones from SYS$SYSTEM:ADA$USE_*ident_string*.COM, where the *ident_string* represents the old version of the compiler. Use the /SYSTEM and /EXECUTIVE qualifiers in the logical name definitions.

When you are using a version of the program library manager (ACS.EXE) that is not the default version and an ACS command creates a command procedure to perform an operation, unexpected results from using a different version of DEC Ada may occur in the following cases:

- If the command procedure is submitted to a batch queue by a /SUBMIT qualifier on the ACS command
- If you later submit the command procedure to a batch queue using the DCL SUBMIT command

Because the program library manager does not transmit knowledge about the current version of the compiler when the command procedure is created, the submitted command procedure (by default) uses the default version of the compiler. For example, if you invoke @SYS$SYSTEM:ADA$USE_INSTALLED.COM and then use an ACS command that creates a command procedure for batch submission, the current Version 3.*n* compiler will be used during the execution of the command procedure.

To override this default, you can modify your LOGIN.COM file so that it invokes the command procedure that redefines the desired compiler and environment.

If you expect to switch frequently between compilers, you may want to modify SYS$STARTUP:ADA$STARTUP.COM or SYS$STARTUP:SYLOGIN.COM to define global OpenVMS symbols to represent the invocations of the command procedure. For example:

```
$ V34 == "@SYS$SYSTEM:ADA$USE_V3_4.COM"
$ V35 == "@SYS$SYSTEM:ADA$USE_INSTALLED.COM"
```

Program libraries generated or modified by a Version 3.5 or higher compiler may not be accepted by older compilers.

During the installation, the system command table (SYS$LIBRARY:DCLTABLES.EXE) is updated to include the new command files associated with the version being installed. Because the changes in the command files are upward compatible, Version 3.*n* commands are accepted by previous versions of the compiler and program library manager. If you use a new Version 3.*n* qualifier or option with a previous version of the compiler, the qualifier or option is ignored.

If this is not acceptable, you can make use of the fact that the command definition file (ADACLD.CLD) associated with the saved compiler is also in the subdirectory that holds the saved compiler. You can use the DCL SET COMMAND command to modify your process's command tables so that the saved compiler's command file is used. For example, if the system command table contains the command definition associated with Version 3.5 but you have redefined your environment so that the Version 3.4 compiler is being used, you can enter the following command to use the Version 3.4 compiler command definition:

```
$ SET COMMAND SYS$COMMON:[SYSEXE.ADA$V3_4]ADACLD.CLD
```

This technique works only as long as you remain in the current process context. When a spawned subprocess is created, the process's current command tables are not copied from the parent process. Instead, the system command table that contains the installed version's command files is used. A spawned subprocess is created when you enter, for example, one of the following commands:

- DCL SPAWN
- LSEDIT COMPILE
- ACS COMPILE
- ACS COMPILE/WAIT
- ACS RECOMPILE
- ACS RECOMPILE/WAIT
- ACS LOAD
- ACS LOAD/WAIT

To reinstall an older version of the compiler, you should perform the installation using the original kit and VMSINSTAL, even if the older version of the compiler has been saved in a subdirectory. Unexpected problems may occur if the files saved in the subdirectory are copied or renamed to the appropriate places.

2.3.1 Saving the Currently Existing Help File

If you chose to keep the currently existing compiler available (in Section 2.1.6), you may also want to save the currently existing Help file. The DEC Ada installation procedure does not automatically save the currently existing Help file. To save the currently existing Help file, you need to extract the file from the system Help library (SYS$LIBRARY:HELPLIB.HLB) and insert it into a newly created user Help library *before* installing the new Help file. For example:

```
$ LIBRARY/EXTRACT/ADA/OUTPUT=USER:[SMITH]V3_nADAHELP.HLP -
_$ SYS$HELP:HELPLIB.HLB
$ LIBRARY/CREATE/HELP USER:[SMITH]V3_nADAHELP.HLB -
_$ USER:[SMITH]V3_nADAHELP.HLP
```

Once you have saved the existing Help file, you can access it by using the /LIBRARY qualifier with the Help command and specifying the user Help library instead of the default system library. For example:

```
$ HELP/LIBRARY=USER:[SMITH]V3_nADAHELP.HLB
```

2.3.2 Saving the Currently Existing Examples Directory

If you chose to keep the currently existing compiler available (in Section 2.1.6), you may also want to save the currently existing examples directory. To save an existing examples directory, you must copy it to another location before installing the new version of DEC Ada.

2.4 Installing DEC Ada on an OpenVMS Cluster

If you want to run DEC Ada on multiple nodes of an OpenVMS Cluster, first check to see that you have the appropriate software license. Then follow these steps after installing DEC Ada:

1. Enter the LICENSE LOAD command to activate the license on each node in the OpenVMS Cluster on which DEC Ada is to be executed as described in the *OpenVMS License Management Utility Manual*.
2. If the other nodes share a common system disk with the installation node, install DEC Ada on the other nodes by running the command procedure SYS$STARTUP:ADA$STARTUP.COM on each node of the cluster that has a software license. You must also update the SYS$LIBRARY:DCLTABLES.EXE file on each of the other nodes by executing the following commands on each node:

   ```
   $ INSTALL
   INSTALL> REPLACE SYS$LIBRARY:DCLTABLES.EXE
   ```

   These commands modify the DCL command tables on the other nodes so that the ADA and ACS commands are recognized and processed.
3. If the other nodes do not share a common system disk with the installation node, you must run a separate installation on each node.

2.5 Postinstallation Notes

DEC Ada is now installed, and the DEC Ada compiler can be invoked by all users with the ADA command. The DEC Ada program library manager can be invoked with ACS commands.

The installation procedure modifies the DCL command table so that the ADA and ACS commands are recognized and processed. However, the previous command table is still in effect for those users who are currently logged in. Users who want to use the ADA or ACS commands must log out of the system and log in again.

Be sure to add the following command to the SYS$MANAGER:SYSTARTUP_VMS.COM file, as requested in Section 2.1.7:

```
@SYS$STARTUP:ADA$STARTUP
```

To ensure device independence when backing up and restoring DEC Ada program libraries, verify that concealed-device logical names have been assigned to all public devices on your system. For information on concealed device logical names, see the *OpenVMS DCL Dictionary* and the *Guide to OpenVMS File Applications*.

For information on using concealed device logical names with DEC Ada program libraries, see *Developing Ada Programs on OpenVMS Systems*.

3
After Installing DEC Ada

This chapter discusses the following tasks, which can take place after installing DEC Ada:

- Running the Installation Verification Procedure separately
- Diagnosing error conditions
- Installing maintenance updates
- Determining and reporting problems

3.1 Running the Installation Verification Procedure Separately

The Installation Verification Procedure (IVP) is usually run at installation. If you want to run the IVP separately to ensure the integrity of installed files should system problems occur, execute the following command procedure:

```
$ @SYS$TEST:ADA$IVP.COM
```

3.2 Error Conditions

If the DEC Ada installation procedure or DEC Ada IVP fails for any reason when you are running VMSINSTAL, the following message is displayed:

```
%VMSINSTAL-E-INSFAIL, The installation of DEC Ada V3.n has failed.
```

An error during the installation can occur if one or more of the following conditions exist:

- Operating system version is incorrect.
- Prerequisite software version is incorrect.
- Quotas necessary for successful installation are insufficient (see Section 1.4.2).
- Process quotas required by VMSINSTAL are incorrect (see Section 1.4.3).

- The OpenVMS Help library is currently in use.
- The product was registered incorrectly or not registered.

For descriptions of the error messages generated by these conditions, see the *OpenVMS System Messages and Recovery Procedures Reference Manual*, your processor-specific installation guide, or the *OpenVMS License Management Utility Manual*.

If you are notified that any of these conditions exist, you should take the appropriate action as described in the message. You may need to change a system parameter or to increase an authorized quota value.

If the installation fails, you must restart the installation procedure from the beginning. If the installation fails due to an IVP failure and the product's License PAK has been registered and loaded, contact your DIGITAL representative.

3.3 Installing Updates

DIGITAL may periodically issue updates of DEC Ada. Each update consists of an installation kit. You should install this kit as described in this document or in any documentation that may accompany the maintenance update.

Each time an update is released, the version number changes. For example, if the current version is 3.2, the version number of the next update may be 3.3. Each update also includes new release notes. The release notes describe the changes that have been made to DEC Ada since the previous release.

The release notes for the updates are provided on line. You should read the release notes when you first install DEC Ada. They are also accessible at any time after the product is installed.

For information on reading the release notes at the time that you install DEC Ada, follow the installation procedure through step 6.

To locate the release notes after DEC Ada is installed, display or print the file SYS$HELP:ADA03*n*.RELEASE_NOTES (where *n* is the number of the current update).

3.4 Determining and Reporting Problems

If you encounter a problem while using DEC Ada, report it to DIGITAL.

Depending on the nature of the problem and the type of support you have, you can call DIGITAL if your software contract or warranty agreement entitles you to telephone support.

Review the Software Product Description (SPD) and Warranty Addendum for an explanation of warranty. If you encounter a problem during the warranty period, report the problem as indicated previously or follow alternate instructions provided by DIGITAL for reporting SPD nonconformance problems.

When you prepare to report a problem, please take the following steps:

1. Describe as accurately as possible the circumstances and state of the system when the problem occurred. Include in the description the version number of DEC Ada being used. Demonstrate the problem with specific examples.
2. Reduce the problem to as small a size as possible.
3. Remember to include listings of any command files, relevant data files, and so on.
4. Provide a listing of the program.
5. If the program is longer than 50 lines, submit a copy of the program on machine-readable media (for example, magnetic tape). If necessary, also submit a copy of the program library used to build the application. Use the OpenVMS Backup utility to copy the program library to the machine-readable media.
6. Report only one problem per report. This facilitates a more rapid response.
7. Mail the problem report to DIGITAL.

Experience shows that many problem reports do not contain enough information to duplicate or identify the problem. Complete and concise information helps DIGITAL give accurate and timely service to software problems.

Also see the Getting Help from DIGITAL section in the Preface.

A
Sample Installation

This appendix contains a sample installation of DEC Ada on an OpenVMS VAX system. This sample was run on a system that had no previous version of DEC Ada installed. Depending on which layered products you have on your system, you might see additional messages and questions when you perform your installation.

The sample installation assumes that DECnet has been shut down, that no users are logged in to your system, and that OPTIONS N is specified to print the release notes. The Installation Verification Procedure (IVP) runs at the end of the installation.

This is only a sample. Sometimes the installation procedure is modified in maintenance updates of this product. Because this installation guide is not usually revised for maintenance updates, be sure to read the *Read Before Installing or Using DEC Ada Version 3.n on OpenVMS VAX Systems* letter before you install this product. This letter notes any significant changes in the installation that vary from this sample.

```
$  @sys$update:vmsinstal ada035 MTA0:

        OpenVMS VAX Software Product Installation Procedure V6.2

It is 20-OCT-1998 at 16:14.

Enter a question mark (?) at any time for help.

* Are you satisfied with the backup of your system disk
[YES]? Return

The following products will be processed:

  ADA V3.5

        Beginning installation of ADA V3.5 at 16:14

%VMSINSTAL-I-RESTORE, Restoring product save set A ...
%VMSINSTAL-I-RELMOVED, Product's release notes have been moved to SYS$HELP.
```

```
   © Digital Equipment Corporation 1998. All rights reserved.

   Restricted Rights: Use, duplication, or disclosure by the U.S.
   Government is subject to restrictions as set forth in subparagraph
   (c) (1) (ii) of DFARS 252.227-7013, or in FAR 52.227-19, or in FAR
   52.227-14 Alt. III, as applicable.

   This software is proprietary to and embodies the confidential
   technology of Digital Equipment Corporation. Possession, use, or
   copying of this software and media is authorized only pursuant to a
   valid written license from Digital or an authorized sublicensor.

* Do you want to run the IVP after the installation [YES]? Return
* Do you want to include SCA analysis data for the predefined
units [YES]? Return
* Do you want to save the currently existing compiler [YES]? N

    There will not be any further questions.

    Installation will take 15 minutes or less, depending on your system
    configuration.

%VMSINSTAL-I-RESTORE, Restoring product save set B ...

    After the installation is complete, you need to add the following
    command to the site-specific system startup command file,
    SYS$MANAGER:SYSTARTUP_VMS.COM;

        $ @SYS$STARTUP:ADA$STARTUP


%VMSINSTAL-I-RESTORE, Restoring product save set C ...
%VMSINSTAL-I-MOVEFILES, Files will now be moved to their target directories...

  Beginning the DEC Ada installation verification procedure.


   © Digital Equipment Corporation 1992, 1998. All rights reserved.

   Restricted Rights: Use, duplication, or disclosure by the U.S.
   Government is subject to restrictions as set forth in subparagraph
   (c) (1) (ii) of DFARS 252.227-7013, or in FAR 52.227-19, or in FAR
   52.227-14 Alt. III, as applicable.

   This software is proprietary to and embodies the confidential
   technology of Digital Equipment Corporation. Possession, use, or
   copying of this software and media is authorized only pursuant to a
   valid written license from Digital or an authorized sublicensor.
```

```
  Please note that this installation verification procedure requires
  that a valid license for DEC Ada has been installed on your system.
   .
   .
   .
%ACS-I-CL_LIBCRE, Library DISK$:[SYS0.SYSUPD.ADA035.ADALIB] created
%ACS-I-CL_LIBIS, Current program library is
        DISK$:[SYS0.SYSUPD.ADA035.ADALIB]
DEC Ada installation verification test
Testing, testing,           1,           2,           3,           4
Testing, testing
DEC Ada installation verification PASSED
%ACS-I-CL_LIBDEL, Library DISK$:[SYS0.SYSUPD.ADA035.ADALIB] deleted

        Installation of ADA V3.5 completed at 16:21

        VMSINSTAL procedure done at 16:21
```

B

Files and Logical Names Installed

The installation procedure installs a number of files on your system and defines some logical names. Section B.1 lists the files installed. Section B.2 lists the logical names that are added to the system logical name table.

B.1 File Names

The installation procedure creates or modifies the following files. However, they do not appear in the installation messages.

SYS$SYSTEM:ADA.EXE	(new)
SYS$SYSTEM:ACS.EXE	(new)
SYS$STARTUP:ADA$STARTUP.COM	(new)
SYS$LIBRARY:DCLTABLES.EXE	(modified)
SYS$SHARE:ADACLD.CLD	(new)
SYS$COMMON:[SYSLIB.ADALIB] and contents	(new)
SYS$COMMON:[SYSLIB]ADA$SCA_PREDEFINED.ANA[1]	(new)
SYS$COMMON:[SYSHLP.EXAMPLES.ADA] and contents	(new)
SYS$HELP:ADA03*n*_SPD	(new)
SYS$HELP:ADA03*n*.RELEASE_NOTES_PS	(new)
SYS$HELP:ADA03*n*.RELEASE_NOTES	(new)
SYS$TEST:ADA$IVP.COM	(new)
SYS$LIBRARY:ADARTL.EXE[2]	(new)
SYS$LIBRARY:ADAMSG.EXE[2]	(new)

[1] Optional.

[2] Not installed if the system already has a copy with a higher version number than provided by this kit.

If you saved the currently existing compiler in Section 2.1.6, the installation procedure also creates the following files:

SYS$COMMON:[SYSEXE.ADA$*ident_string*[1]] and contents	(new)
SYS$SYSTEM:ADA$USE_*ident_string*[1].COM	(new)
SYS$SYSTEM:ADA$USE_INSTALLED.COM	(new)

[1] *ident_string* represents the version number of the currently installed DEC Ada compiler.

B.2 Logical Names

Table B–1 lists the logical names that are entered into the system logical name table when you install DEC Ada. These names are stored in the product's startup file. They are automatically entered into the system logical name table whenever the system reboots or whenever the software is invoked.

Table B–1 DEC Ada Logical Names and Locations

Logical Name	Location
ADA$EXAMPLES	SYS$COMMON:[SYSHLP.EXAMPLES.ADA]
ADA$HTML	DEC Ada manuals in .HTML format SYS$COMMON:[SYSHLP.EXAMPLES.ADA.HTML]
ADA$PREDEFINED	ADA$PREDEFINED_ROOT:[ADALIB]
ADA$PREDEFINED_ROOT	Defined as a concealed device logical name that is equivalent to SYS$COMMON:[SYSLIB.]

C
Tuning OpenVMS VAX Systems

When configuring a system to handle DEC Ada compilations, you need to consider all of the activities that you wish to perform. For example, you need to consider the amount of parallel compilation that the system can support as well as considering the amount of memory you need.

In general, a process used for compiling DEC Ada programs should have a WSEXTENT quota of at least 8,000. If you are performing compilations interactively, this quota should be specified as a SYSUAF quota.

For compilations done in a batch queue, it is possible for the system manager to set the WSEXTENT attribute for the batch queue. In that case, the system should be configured with enough memory to let the maximum desired number of concurrent compilations reach their maximum working set simultaneously. Otherwise, the system may start paging or swapping excessively.

C.1 General Tuning of OpenVMS Systems

An optional DIGITAL product, the Disk File Optimizer, can be used to reorder fragmented files online without requiring any system downtime or affecting disk availability. This product can also migrate files to their optimal places on a disk.

C.2 Improving Product Performance

This section gives detailed information on how to improve the performance of the DEC Ada compiler and program library manager. Every effort was made to provide the appropriate parameters for tuning your system. However, you may gain better performance by adjusting some of the parameters upwards or downwards to suit your particular system configuration.

Also in this section, System Generation utility (SYSGEN) and user authorization file (UAF) parameters are described, and suggested minimum values are noted. The parameters discussed are those for which DEC Ada requires larger values than the OpenVMS defaults or for which the values are larger than those required for other OpenVMS programming languages.

The OpenVMS operating system controls resource usage through two primary sets of parameters: systemwide parameters and per-process parameters. You can use the SYSGEN utility to define and modify systemwide parameters (SYSGEN parameters). Per-process parameters are controlled on a per-user-name basis by quotas and defaults that are contained in the user authorization file (UAF parameters). You use the Authorize utility to modify UAF entries.

A complete description of OpenVMS resources, SYSGEN, and UAF parameters is contained in the *OpenVMS System Manager's Manual*. You should be familiar with the procedures described there for modifying SYSGEN parameters.

The recommended procedure is to add site-specific parameters to the file SYS$SYSTEM:MODPARAMS.DAT and then invoke the SYS$UPDATE:AUTOGEN command procedure. The AUTOGEN command procedure computes the optimal value for all parameters. See the *OpenVMS System Manager's Manual* for information on how to use AUTOGEN and how to specify parameter values before invoking AUTOGEN.

DIGITAL recommends that you run AUTOGEN with feedback about one week after installing DEC Ada or after the DEC Ada load changes significantly. Whenever possible, modify the MODPARAMS.DAT file to include MIN_ *parameter* or ADD_*parameter* instead of hard-coding values into this file.

C.2.1 ASTLM

The UAF AST queue limit (ASTLM) parameter limits the sum of the following:

- The number of asynchronous system trap (AST) requests that a user's process can have outstanding at one time
- The number of scheduled wakeup requests that a user's process can have outstanding at one time

The DEC Ada delay statement is implemented as a call to the OpenVMS SYS$SETIMR system service, which executes an AST routine. DEC Ada also provides facilities for calling OpenVMS system routines that execute AST routines. For example, the package STARLET provides the system services SYS$QIO and SYS$QIOW both of which let you specify an AST service routine. The routines in the package TASKING_SERVICES execute ASTs as well (although the execution and handling of the ASTs is hidden by the package).

The suggested OpenVMS value for ASTLM is 24. This should be sufficient for most DEC Ada programs. If your programs involve tasks that could execute many delay statements simultaneously or if you use a high number of calls

to OpenVMS system routines that execute AST routines, you may want to increase this value.

See the *DEC Ada Run-Time Reference Manual for OpenVMS Systems* for more information on ASTs in DEC Ada programs, especially programs that use tasks.

C.2.2 ENQLM

The UAF enqueue quota (ENQLM) parameter limits the number of locks that a process and its subprocesses can own. OpenVMS Record Management Services (RMS) uses locks to synchronize access to shared files and records.

The DEC Ada compiler and program library manager use RMS to access the DEC Ada program library index file, ADALIB.ALB. Up to eight locks can be used for each library or sublibrary that is open. When a sublibrary is opened, usually its parent and its ancestors are also opened. If you are using a sublibrary that has a parent and a grandparent, up to 24 (3 * 8) locks may be needed. Commands such as ACS COPY UNIT can have two libraries or sublibraries open at the same time, which doubles the number of locks.

Note

The suggested OpenVMS value for ENQLM is 30. This value is not sufficient for using the DEC Ada program library manager and sublibraries. The recommended value for DEC Ada users is 60.

C.2.3 FILLM

The UAF open file limit (FILLM) parameter limits the number of files that a user's process can have open at one time. This limit includes the number of network logical links that can be active at the same time.

The DEC Ada compiler, the DEC Ada program library manager, and the OpenVMS Linker use the value of FILLM to limit the total number of files open at one time. If a compilation involves a large number of units and FILLM is set too low to allow all the files involved to be opened at the same time, the compilation can take a long time. For example, if a unit depends directly or indirectly on 99 additional units and if FILLM is set to 20, the compilation is slower than if FILLM had been set to 100.

FILLM is a pooled limit with a suggested value of 20. This value is not sufficient for most DEC Ada compilations. The recommended minimum value for DEC Ada users is 50. The value of FILLM must always be lower than the value of the SYSGEN channel count (CHANNELCNT) parameter.

If you increase the value of the FILLM parameter, you may also want to increase the value of the UAF BYTLM parameter. The general rule for the relationship between these two parameters is that the value of BYTLM should be at least 100 times the value of FILLM.

C.2.4 PRCLM

The UAF subprocess creation limit (PRCLM) parameter limits the number of subprocesses that a user's process can create.

The DEC Ada program library manager creates a subprocess to run the linker or the DEC Ada compiler for the ACS LINK/WAIT, COMPILE/WAIT, LOAD/WAIT, and RECOMPILE/WAIT commands. Only one subprocess is created, and the creating process waits for the termination of the subprocess.

The suggested value for PRCLM is 2. This should be sufficient for the DEC Ada program library manager as well.

C.2.5 TQELM

The UAF timer queue entry limit (TQELM) parameter limits the sum of the following:

- The number of entries that a user's process can have in the timer queue
- The number of temporary common flag clusters that a user's process can have

The DEC Ada delay statement is implemented as a call to the OpenVMS SYS$SETIMR system service, which adds an entry to the timer queue.

The suggested OpenVMS value for TQELM is 20. This value should be sufficient for most Ada programs. If your programs involve tasks that could execute many delay statements simultaneously, you may want to increase this value.

See the *DEC Ada Run-Time Reference Manual for OpenVMS Systems* for more information on the interaction of the TQELM parameter with DEC Ada tasking programs.

C.2.6 Virtual Memory Usage

Various SYSGEN and user parameters should be set so that a DEC Ada compilation can use up to 20,000 pages (10 megabytes) of virtual memory. The following sections describe working set parameters that should be adjusted to provide good performance for such large virtual address spaces.

- VIRTUALPAGECNT parameter

The SYSGEN VIRTUALPAGECNT parameter sets the maximum number of virtual pages that can be mapped for any one process. The OpenVMS default (65,536 pages, or 32 megabytes) should be adequate for compilations of most DEC Ada programs.

- PGFLQUOTA parameter

  The UAF paging file quota (PGFLQUOTA) parameter limits the number of pages that your process can use in the system paging file. It limits the amount of read/write working storage that the compiler can use. PGFLQUOTA should be set to a value consistent with VIRTUALPAGECNT.

  The suggested OpenVMS value of 12,800 for PGFLQUOTA is too low. A value of at least 17,000 is recommended. It is unnecessary to raise the value of PGFLQUOTA above (VIRTUALPAGECNT—3,000).

  If the value of PGFLQUOTA or the system's value of VIRTUALPAGECNT is too low, large DEC Ada compilations fail and you see the following message:

  ```
  %F, Insufficient virtual memory
  ```

- System Paging File

  The system paging file (SYS$SYSTEM:PAGEFILE.SYS) determines the amount of paging space available for system processes and RMS global buffers.

  The paging file size should be greater than the maximum paging file quota (UAF PGFLQUOTA parameter) for an individual. It should also be greater than the virtual page count (SYSGEN VIRTUALPAGECNT parameter). For example, if the value of VIRTUALPAGECNT is 20000 pages, then a paging file of at least 30000 blocks is needed.

  If the paging file size is not properly adjusted with respect to these two parameters, system processing could appear to be suspended and the following messages is displayed:

  ```
  %SYSTEM-W-PAGEFRAG, Page file badly fragmented, system continuing
  %SYSTEM-W-PAGECRIT, Page file space critical, system trying
   to continue
  ```

  It is recommended that you use the DIGITAL-supplied AUTOGEN command procedure to suggest the size of the system paging file after adjusting the values of VIRTUALPAGECNT and PGFLQUOTA. See Section 1.4.2.3, or the *OpenVMS System Manager's Manual* for information on how to use AUTOGEN and how to specify parameter values before invoking AUTOGEN.

See the *OpenVMS System Manager's Manual* for information on the OpenVMS default value of PGFLQUOTA. See the *VMS System Generation Utility Manual* for more information on the OpenVMS default value of VIRTUALPAGECNT.

- WSQUOTA and WSEXTENT parameters

  The UAF working set quota (WSQUOTA) parameter specifies the maximum size to which a user's physical memory size can grow on a typically loaded system. The system guarantees the user that WSQUOTA physical pages are available to the user's process.

  The UAF working set extent (WSEXTENT) parameter specifies the maximum size to which a user's physical memory can grow, independent of system load. WSEXTENT should be greater than or equal to WSQUOTA.

  If WSEXTENT is greater than WSQUOTA, the OpenVMS operating system attempts to provide additional physical pages to a process that is page-faulting heavily. On a lightly loaded system, the user's working set can grow beyond WSQUOTA up to WSEXTENT.

  A DEC Ada compilation should have a working set of 2500 pages or more. If a user is going to do DEC Ada compilations interactively or in a subprocess (COMPILE/WAIT), the user's UAF entry should specify WSQUOTA=2500.

  Many sites may not permit large working sets for interactive jobs. In this case, a batch queue should be established for DEC Ada compilations. The DEC Ada batch queue can define values for WSQUOTA and WSEXTENT that override the UAF values. Compilations done in the batch queue can have much larger working sets, resulting in better compile times and better use of system resources.

C.2.7 Batch Queue Parameters

A batch queue with large working set parameters should be provided for DEC Ada compilations. This batch queue minimizes the number of concurrent DEC Ada compilations so that each compilation completes efficiently without inducing excessive system overhead.

You should define the logical name ADA$BATCH in the system logical name table. The DEC Ada program library manager submits ACS COMPILE and RECOMPILE commands to the queue named by ADA$BATCH. If ADA$BATCH is not defined, these jobs are submitted to SYS$BATCH.

A typical DEC Ada batch queue that is designed to handle two concurrent compilations should be initialized with the following values:

```
$INITIALIZE/QUEUE/BATCH-
/BASE_PRIORITY=4/JOB_LIMIT=2-
/WSQUOTA=2500/WSEXTENT=4000     ADA_BATCH

$ ASSIGN/SYSTEM ADA_BATCH ADA$BATCH
```

C.2.8 WSMAX Parameter

The SYSGEN working set maximum number of pages (WSMAX) parameter sets the maximum number of pages for any working set on a systemwide basis. The value of WSMAX should be as great as the maximum WSEXTENT value assigned to any user or batch queue.

The OpenVMS default WSMAX of 1024 pages is too low for DEC Ada. A value of at least 4000 is recommended.

C.2.9 Channel Count Parameters

The SYSGEN channel count (CHANNELCNT) parameter specifies the maximum number of channels for each process in an OpenVMS system. The UAF file and logical link limit (FILLM) parameter specifies the maximum number of files that can be open at one time, including active network logical links.

The DEC Ada compiler, DEC Ada program library manager, and the OpenVMS Linker use the value of FILLM to limit the total number of open files at one time. They close files as necessary to avoid exceeding that value.

The compiler and program library manager assume that the value of FILLM (which has a default value of 20) is less than the value of CHANNELCNT (which has a default value of 127). You may need to raise the value of FILLM to accommodate DEC Ada compilations that involve a large number of compilation units. You should be careful not to raise the value of FILLM above the value of CHANNELCNT.

C.2.10 Program Library Networking Effects

There are several SYSGEN and DECnet parameters (on both the local and remote nodes) that can affect the availability of compilation units or files accessed over DECnet. For example, every time a file is opened on a remote node, a temporary connection, called a logical link, is made from the local node to the remote node. The total number of logical links allowed at one time is controlled by DECnet and can be set by the Network Control Program (NCP) utility with the following command:

```
NCP> SET EXECUTOR MAX LINKS X
```

The *X* in this command is the maximum number of logical links. This number represents the system (not process) quota. Each connection between two nodes deducts one from the quota total. When setting this value, note that both the DEC Ada compiler and program library manager use the per-process FILLM (file and logical link limit) quota, not the system quota, to limit the total number of open files at one time.

Limiting the total number of files open at one time also reduces the potential number of logical links created to the remote node. A logical link is required only for files that are not accessed locally.

Because the creation of a logical link may also involve the creation of a process on the remote node, you may need to increase (or at least monitor) the values of the SYSGEN parameters MAXPROCESSCNT and BALSETCNT. This allows more processes to be created for libraries for network servers (FALs).

If you are expecting to access a number of files or compilation units over DECnet, you may want to increase the value of the UAF buffered input-output byte count limit (BYTLM) parameter on your system. The value of this parameter affects the efficiency of program library operations performed over DECnet. Other parameters may also have an effect. DECnet parameters are documented in the *DECnet for OpenVMS Guide to Networking*.

C.3 Relevant File Quotas and Cache Sizes

The following sections describe how to estimate the sizes that the various OpenVMS caches should be set to.

The relevant system manuals should be consulted to determine whether the resulting cache sizes are supportable (perhaps after installing more physical memory).

The major classes of data that must be read from the disks are as follows:

- The files themselves, especially the ADALIB.ALB and the .ACU files
- The file headers from within INDEXF.SYS that describe where the files themselves reside
- The directory files that describe where the file headers reside, given a file name

C.3.1 Caching Files

Each time DEC Ada performs a compilation or ACS operation, the following disk files are cached into memory as needed:

- DEC Ada compiler
- DEC Ada program library manager
- Contents of all necessary .ACU files

For both small and large programs, caching can save time when performing compilations or ACS operations by avoiding unnecessary disk accesses. You can maximize the benefits of caching by:

- Invoking the program library manager interactively. For example:

```
$ ACS
ACS>
```

- Specifying more than one unit with the DCL ADA or ACS command, whenever possible. For example:

```
ACS> COMPILE UNITA, UNITB, UNITC
```

C.3.2 Caching File Headers

The OpenVMS system uses file header caches to greatly reduce the time needed to open files. The size of the cache is controlled by the SYSGEN parameter ACP_HDRCACHE. OpenVMS sets a default value of 128 pages for ACP_HDRCACHE, but this value may be too small for the typical DEC Ada program running on your system.

Because of the way the DCL ADA and ACS commands access files, it is important that the value of ACP_HDRCACHE is large enough to keep all the accessed file headers cached. If the cache is not large enough, the potential performance benefits of the cache are generally not realized.

To estimate the value needed for ACP_HDRCACHE, you need to sum the total number of files that a typical DEC Ada program may require to compile, link, and run successfully. The following lists the files that you need when estimating a value for ACP_HDRCACHE:

- Total number of .ACU files in your program library
- Total number of .OBJ files in your program library
- Total number of .ADA source files in your program
- Total number of .ALB files in your program library

- Total number of directories that must be accessed, including any subdirectories that are accessed
- Total number of system files that your program accesses
- Total number of other files that your program accesses
- Some for other users on the system

You must include units from the predefined library in your estimates. You do not need to add the total number of .ADC files (copied source files) to your estimate because they are typically not accessed.

Suppose the typical DEC Ada program on your system is as follows:

- Contains 220 .ACU files, 220 .OBJ files, and 100 .ADA files. (For this example, the predefined units are considered part of the 220 .ACU files.)
- Located in two sublibraries and one parent library (making for a total of 3 .ALB files).
- Accesses 5 directories.
- Accesses 15 system files.
- Accesses 8 other files.

The following example shows the calculation for the preceding program:

```
    220    .ACU files
    220    .OBJ files
    100    .ADA files
      3    .ALB files
      5    Directories
     15    System files
 +    8    Other files
-----------------------
    571    Estimate for ACP_HDRCACHE
```

It is not unusual to have an estimated value for ACP_HDRCACHE that is over 500.

One way to increase the value of ACP_HDRCACHE is to modify the MODPARAMS.DAT file and run AUTOGEN. For example, suppose you wanted to increase the value of ACP_HDRCACHE by more than the amount you estimated. To do this, enter the following line in the MODPARAMS.DAT file:

```
ADD_ACP_HDRCACHE = 600
```

For the new value of ACP_HDRCACHE to take effect, you must run AUTOGEN. If you run AUTOGEN with feedback, it increases the size of the file header cache in small increments if the header cache is too small. However when the system load can be predicted, it is better to modify the MODPARAMS.DAT file than to use AUTOGEN with feedback.

C.3.3 Caching Directory Files

In general, the program library manager accesses directory files in a random order. Consequently, the ACP_DIRCACHE parameter should be set to the sum of the sizes of all the directories needed to be accessed when compiling and linking your programs. Unlike the ACP_HDRCACHE, this number gives steady improvement as it gets better, rather than a dramatic improvement as it crosses a threshold.

C.4 Using DECram on OpenVMS Systems

Using pseudo-disks, such as DECram, can be an effective way of reducing disk traffic. For example, you can locate your current program library and any other frequently accessed libraries on the DECram disk.

You can also use a DECram disk to place the copy (use ACS COPY) of a library so that you can access those files quickly. Once you create a copy of the units in the new library, any updates to the original library can cause the new library and the original library to be out of synchronization.

If you do not have enough space on your DECram disk for all the units in your DEC Ada program, consider placing only the specifications of the units on the DECram disk. Most of the units that are accessed during a compilation are specifications.

Note

DECram for OpenVMS does not support long-term data storage because data in a DECram device is volatile.