<comment>(changed next tag for CUP_HANDBOOK)

VAX Ada Installation Guide

AA–EF85C–TE

May 1989

This document contains instructions for installing VAX Ada on the VMS operating system. It also explains how to read the online release notes before or after installing the product. This document applies to VAX Ada Version 2.0 and all maintenance updates throughout that version.

Revision/Update Information: This revised manual supersedes *Installing VAX Ada* (Order Number AA–EF85B–TE)

Operating System and Version: VMS Version 5.0 or higher. Future maintenance releases may require higher versions.

Software Version: VAX Ada Version 2.0

digital equipment corporation
maynard, massachusetts

First Printing, February 1985
Revised, May 1987
Revised, May 1989

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

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

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

© Digital Equipment Corporation 1985, 1987, 1989.

All Rights Reserved.
Printed in U.S.A.

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

ALL–IN–1	EduSystem	RT
DEC	IAS	ULTRIX
DEC/CMS	MASSBUS	UNIBUS
DEC/MMS	PDP	VAX
DECnet	PDT	VAXcluster
DECmate	P/OS	VMS
DECsystem–10	Professional	VT
DECSYSTEM–20	Q–bus	Work Processor
DECUS	Rainbow	VAXELN
DECwriter	RSTS	
DIBOL	RSX	digital™

ZK3295

This document was prepared using VAX DOCUMENT, Version 1.1

Contents

Preface v

Preparing to Install VAX Ada 1
Prerequisite Software 1
Accessing the Online Release Notes 1
Installation Procedure Requirements 1
User Account Quotas and Privileges 3
VMS License Management Facility 4
Using VMSINSTAL 5
Installing VAX Ada 6
Keeping the Currently Existing Compiler Available 13
Installing VAX Ada on a VAXcluster 15
Installing the VAX Ada HELP File 16
After the Installation 18
Running the Installation Verification Procedure Separately 19
Error Conditions 19
Installing VAX Ada as a Shared Image 20
Maintenance Updates 20
Improving Product Performance 21
Memory Usage 21
Working Sets 21
Effect of Working Set on Paging Rate 23
Effect of Working Set on Compilation Rate 23
Suggestions for Controlling Working Set Sizes 25
Virtual Address Space 26
Resource Requirements 27
ASTLM—AST Queue Limit Parameter 27
ENQLM—Enqueue Quota Parameter 28
FILLM—Open File Limit Parameter 28
PRCLM—Subprocess Creation Limit Parameter 29
TQELM—Timer Queue Entry Limit Parameter 29

Virtual Memory Usage 29
VIRTUALPAGECNT—Maximum Number of Virtual Pages Parameter 30
PGFLQUOTA—Paging File Quota Parameter 30
System Paging File 30
WSQUOTA and WSEXTENT—Working Set Quota and Extent Parameters 31
Batch Queue Parameters 31
WSMAX—Working Set Maximum Number of Pages Parameter 32
Program Library Networking Effects 32
Channel Count Parameters 33
Determining and Reporting Problems 33
Sample Installation 34

Figures

1 Page Faults Versus Working Set Size 24
2 Compilation Rate Versus Working Set Size 25

Tables

1 Description of Test Programs 22

Preface

This document describes how to install VAX Ada on the VMS operating system. It is intended for system managers. This document applies to VAX Ada Version 2.0 and all maintenance updates in that version. A major version number is represented by the digits to the left of the decimal point, and the maintenance update version number is indicated by the digits to the right of the decimal point.

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

At the back of this document are two Reader's Comments forms. If you have any comments about this document, fill out one form and mail it; your comments will be considered for the next revision of the installation guide.

For more information about installing VAX Ada on VMS systems, see your processor-specific installation/operations guide.

The installation procedure is described step-by-step and is, for the most part, self-explanatory. The system prompts you to answer questions and to change volumes, and waits for you to tell it to continue.

Defaults appear in brackets ([ ]) throughout the installation procedure. Text you type is indicated in red.

Preparing to Install VAX Ada

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

Prerequisite Software

VAX Ada requires Version 5.0 or higher of the VMS operating system.

The VMS operating system comes with a variety of support options, or classes. Classes include such features as networking and RMS journaling. To use VAX Ada, your system should be running a version of VMS that includes the classes listed in the Software Support Addendum (SSA), which comes with the Software Product Description (SPD).

Accessing the Online Release Notes

VAX Ada provides online release notes. The prompt to display or print the release notes during the installation appears in step 4 of the installation procedure when you invoke VMSINSTAL with the OPTIONS N parameter.

To obtain the release notes after the installation, type or print the file SYS$HELP:ADA02 *n*.RELEASE_NOTES. The *n* represents the maintenance update of VAX Ada covered in the release notes. Thus, in the case of VAX Ada Version 2.0, the release notes are in the file SYS$HELP:ADA020.RELEASE_NOTES.

The VAX Ada HELP file also contains the release notes for the current version of VAX Ada, as well as the accumulated release notes and documentation changes and corrections for maintenance updates. After you have installed the HELP file, type the following:

```
$ HELP ADA RELEASE_NOTES RETURN
```

If your system had a previous version of VAX Ada installed, you should check the release notes for information on any program library changes.

Installation Procedure Requirements

The installation takes approximately 25 minutes, depending on the type of media and your system configuration. Before installing VAX Ada, you must have the following privileges and resources:

- SETPRV privileges, or CMKRNL, WORLD, and SYSPRV privileges
- 33,000 blocks of free disk space, minimum, during installation for the VAX Ada compiler and associated components
- 2,000 blocks of free disk space, minimum, during installation for the VAX Ada HELP file

- If VAX Ada already exists on your system and either the compiler (ADA.EXE) or program library manager (ACS.EXE) was specified as a known or shared image, VMSINSTAL automatically attempts to install the new versions as known or shared images. In this case, the installation will require the following number of global sections and global pages:

Image	Global Sections	Global Pages
ADA.EXE	5	3000
ACS.EXE	3	500

  Note that if you are running VAX Ada on a VAXcluster, 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 VMS Install Utility.

  See the Installing VAX Ada as a Shared Image section for more information.

To determine the number of available global pages on your system, invoke the VMS Install Utility by typing the following commands:

```
$ INSTALL :== $INSTALL/COMMAND_MODE RETURN
$ INSTALL RETURN
```

Then type the following command:

```
INSTALL> LIST/GLOBAL/SUMMARY RETURN

        Summary of Local Memory Global Sections

    379 Global Sections Used,  31922/39978 Global Pages Used/Unused

INSTALL> EXIT RETURN
```

When you enter the LIST/GLOBAL/SUMMARY command, the system displays a summary of the number of global sections used, the number of global pages used, and the number of global pages unused. To exit from the VMS Install Utility, type the EXIT command.

Note *If the necessary number of global pages and global sections is not available, the DCL tables (SYS$LIBRARY:DCLTABLES.EXE) could be corrupted in some situations.*

If there are fewer than the required number of global pages available, you can delete an existing known image with the VMS Install Utility, or you can increase the GBLPAGES system parameter by editing the file SYS$SYSTEM:MODPARAMS.DAT and then invoking the SYS$UPDATE:AUTOGEN command procedure. For more information on

modifying system parameters and using AUTOGEN, see the *Guide to Setting Up a VMS System*.

You can determine the number of available global sections by invoking SYSGEN as follows:

```
$ RUN SYS$SYSTEM:SYSGEN RETURN
SYSGEN> USE CURRENT RETURN
SYSGEN> SHOW GBLSECTIONS RETURN
    .
    .
    .
SYSGEN> EXIT RETURN
```

Compare the maximum of global sections with the number of global sections in use displayed by the VMS Install Utility. If the number of global sections used plus the number of global sections required for VAX Ada exceeds the maximum number of global sections, you must increase the number of available global sections before you can install VAX Ada as a shared image. To increase the number, you must increase the GBLSECTIONS system parameter by editing the file SYS$SYSTEM:MODPARAMS.DAT and then invoking the SYS$UPDATE:AUTOGEN command procedure. For more information on modifying system parameters and using AUTOGEN, see the *Guide to Setting Up a VMS System*.

If the number of global sections used plus the number of global sections required for VAX Ada is less than the maximum number of global sections, you can exit from SYSGEN and continue.

User Account Quotas and Privileges

To use VAX Ada, each account or batch queue that will be invoking the VAX Ada compiler must have TMPMBX and NETMBX privileges and the following requirements:

- AST queue limit (ASTLM): 24 (minimum)
- Enqueue quota (ENQLM): 60 (minimum)
- Open file quota (FILLM): 50 (minimum)
- Paging file quota (PGFLQUOTA): 17000 pages (minimum)
- Subprocess creation quota (PRCLM): 2 (minimum)
- Timer Queue Entry Limit (TQELM): 20 (minimum)
- Working set size (WSDEF): 750 pages (minimum)
- Working set quota (WSQUOTA): 2500 pages (minimum)
- Working set extent (WSEXTENT): 4000 pages (minimum)

The Resource Requirements section (in the Improving Product Performance section) discusses these, as well as other, resources in more detail.

Use the VMS Authorize Utility (AUTHORIZE) to compare the current values of these quotas with the requirements for VAX Ada. The account used in the following example is the SYSTEM account:

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

To change the values of these quotas, use the VMS Authorize Utility's MODIFY command. For more information, see the *VMS Authorize Utility Manual*.

VMS License Management Facility

The VMS License Management Facility (LMF) is available with Version 5.0 of the VMS operating system. Before installing VAX Ada, you must register and load your VAX Ada software license.

The license registration information you need is contained in the Product Authorization Key (PAK) that is shipped with VAX Ada. The PAK is a paper certificate that contains information about the license you have to run a particular piece of software.

During the installation, you are asked if you have registered the VAX Ada license and loaded the appropriate authorization key. You must register and load your license for VAX Ada *before* you start the installation.

To register a license under VMS Version 5.0, first log in to a privileged account. Digital recommends that you register and load the license from the system manager's account. 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 Product Authorization Key (PAK).
- Enter the DCL LICENSE REGISTER command with the appropriate qualifiers that correspond to information on the PAK.

Depending on the way in which you registered the VAX Ada license, you also have a choice of two ways to load the license.

- Type YES in response to the appropriate question when you are running SYS$UPDATE:VMSLICENSE.COM.
- Enter the DCL LICENSE LOAD command.

Note that you need the CMKRNL, SYSNAM, and SYSPRV privileges to load a license. For complete information on using LMF, see the manual on the License Management Utility in the VMS documentation set.

Using VMSINSTAL

After you log in to a privileged account to install VAX Ada, invoke VMSINSTAL, the command procedure that installs this product. An example of this procedure is shown in steps 1 and 2 of the Installing VAX Ada section.

The VMSINSTAL command procedure has the following syntax:

```
$ @SYS$UPDATE:VMSINSTAL ADA02n ddcu: [OPTIONS N]
```

n

Represents the maintenance update of VAX Ada. Refer to the label on the VAX Ada distribution kit or the bill of materials for the maintenance update number of your kit. For VAX Ada Version 2.0, the maintenance update is 0.

You can also type an asterisk (*) in place of ADA02*n*. If you are installing from magnetic tape, you can use the asterisk to save installation time; the asterisk causes both the compiler kit and HELP kit to be installed with a minimum number of tape passes.

For example, the first parameter to VMSINSTAL for VAX Ada Version 2.0 could be either ADA020 or *.

ddcu:

Represents a device name where the distribution volumes will be mounted for the VAX Ada installation media, where *dd* is the device code, *c* is the controller code, and *u* is the unit number. It is not necessary to use the console drive to install VAX Ada. If you do use the console drive, you should replace any media you removed from the drive. MUA0: is the device name used in the examples in this document.

OPTIONS N

Is an optional parameter you should provide if you want to be prompted to display or print the release notes. If you do not include the OPTIONS N parameter, VMSINSTAL does not prompt you to display or print the release notes. VMSINSTAL permits the use of several other options. For more information on the other options, see your processor-specific installation/operations guide.

VMSINSTAL requires the product and device names. If you do not supply these parameters, VMSINSTAL prompts you for them. VMSINSTAL will also prompt you for the optional parameter that indicates whether or not you want the release notes displayed or printed. If you wish to be prompted for the product and device names and to display or print the release notes, type the following:

```
$ @SYS$UPDATE:VMSINSTAL RETURN
```

When you invoke VMSINSTAL, it checks the following:

- Whether you logged in to a privileged account. Digital recommends that you install software from the system manager's account with your default device and directory set to SYS$UPDATE.
- Whether you have adequate quotas for installation. VMSINSTAL checks for the following quota values:
  - ASTLM = 24
  - BIOLM = 18
  - BYTLM = 18000
  - DIOLM = 18
  - ENQLM = 30
  - FILLM = 20

  VMSINSTAL then checks whether the following conditions exist. If so, discontinue the conditions before resuming the installation procedure.

  - Whether DECnet is up and running
  - Whether any users are logged in to the system

  If VMSINSTAL detects either of these conditions, you will be asked whether you want to continue the installation. If you want to continue, type YES. If you want to stop the installation, press RETURN.

Installing VAX Ada

This section contains excerpts from the installation procedure and explains each step. Defaults appear in brackets throughout the installation procedure. Note that the VAX Ada HELP file is a separate kit. The procedure for installing that kit is documented in the Installing the VAX Ada HELP File section.

To abort the installation procedure at any time, press CTRL/Y. When you press CTRL/Y, the installation procedure deletes all files it has created up to that point and then returns you to DCL level. If you want to retry the installation procedure after pressing CTRL/Y, you must proceed from step 2.

Step 1: Log in to a privileged account and set your default device and directory to SYS$UPDATE.

```
RETURN
Username: SYSTEM   RETURN
Password:          RETURN
$ SET DEFAULT SYS$UPDATE  RETURN
```

Step 2: Invoke VMSINSTAL.

```
$ @VMSINSTAL ADA02n MUA0: OPTIONS N RETURN

        VAX/VMS Software Product Installation Procedure VX.X


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

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

If you are satisfied with the backup of your system disk, press RETURN. Otherwise, type NO and press RETURN to discontinue the installation. Digital recommends that you back up your system disk before installing any software.

Step 3: Insert the installation kit volume.

```
Please mount the first volume of the set on MUA0:.
* Are you ready? YES RETURN
The following products will be processed:

  ADA V2.n


        Beginning installation of ADA V2.n at hh:mm

%VMSINSTAL-I-RESTORE, Restoring product saveset A...
```

To indicate that you want to continue the installation, mount volume 1, type YES, and press RETURN.

Step 4: Select a release notes option.

This step applies only if you specified OPTIONS N in step 2.

```
Release Notes Options:

    1. Display release notes
    2. Print release notes
    3. Both 1 and 2
    4. Copy release notes to SYS$HELP
    5. Do not display, print or copy release notes

* Select option [2]: RETURN
```

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

If you select option 2, VMSINSTAL prompts you for a queue name. Either type a queue name or press RETURN to send the file to the default output print device.

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

If you select option 3, VMSINSTAL displays the release notes on line immediately. You can terminate the display at any time by pressing CTRL/C. VMSINSTAL then prompts you for a queue name. Either type a queue name or press RETURN to send the file to the default output print device. The VAX Ada release notes are then copied to SYS$HELP:ADA02*n*.RELEASE_NOTES.

If you select option 4, VMSINSTAL copies the release notes to SYS$HELP immediately.

If you select option 5, VMSINSTAL does not display, print, or copy the release notes; typically you select this option when you want to terminate the installation.

Note *This version of the release notes file installed by VMSINSTAL is labeled with the current product name and version number. Take care not to delete release notes for previous versions of VAX Ada.*

Next, VMSINSTAL displays the following prompt:

```
* Do you want to continue the installation?: Y RETURN
%VMSINSTAL-I-RELMOVED, The product release notes have been successfully
 moved to SYS$HELP.
```

If you want to continue the installation, type YES and press RETURN. If you type NO or press RETURN, VMSINSTAL discontinues the installation. In either case, VMSINSTAL moves the release notes to SYS$HELP.

Step 5: Select installation options.

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

The Installation Verification Procedure (IVP) runs tests to check whether the installation procedure was successful. Press RETURN to run the IVP after the installation. If you do not want to run the IVP, type NO in response to the prompt.

```
* Do you want to include the SCA library for the predefined
  units [YES]? RETURN
```

VAX Ada provides support for the VAX 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, subprograms, and so on contained in the predefined units.

If you have SCA installed on your system and intend to use it to analyze 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, then type NO in response to the prompt.

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

```
* Do you want to include the sources for the predefined units [YES]? RETURN
```

The copied source files for the VAX Ada predefined units (.ADC files for the units STARLET, TEXT_IO, VAXELN_SERVICES, and so on) are included automatically if you typed YES in response to the previous prompt about the predefined SCA library. If you typed NO in response to the prompt about the SCA library for the predefined units, and you want to have the copied source files for the VAX Ada predefined units included in the library of predefined units (ADA$PREDEFINED), type YES or press the RETURN key in response to this prompt.

You need the copied source files if you want to use the ACS EXTRACT SOURCE command with these units or are otherwise interested in the source code for them. If you are concerned about saving disk space, and do not have a need for the copied source files, you can type NO in response to this prompt.

If you need to know how much disk space these files will take before you can answer the prompt, then type a question mark (?) to obtain specific information on disk space requirements before typing YES or NO.

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

If the current version of VAX 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 need to know exactly which files are saved, then type a question mark (?) to obtain specific information before typing YES or NO.

See the Keeping the Currently Existing Compiler Available section for more information.

Step 6: Respond to License Registration Queries.

VAX Ada supports the VMS License Management Facility (LMF). The installation procedure displays license information about your product and then asks if you have registered and loaded your authorization key for VAX Ada. The following is an example of such information:

```
        Product:        ADA
        Producer:       DEC
        Version:        2.n
        Release Date:   dd-mmm-yyyy

* Does this product have an authorization key registered and
  loaded? Y RETURN
```

If you have not registered and loaded your authorization key, you must answer ''NO'' to this question. You must register and load your product authorization key to successfully complete the installation. If you have not done so, you should stop the installation, register and load your product authorization key, and then restart the installation.

Step 7: Read informational messages.

The VAX Ada installation procedure produces a number of informational messages that report on the progress of the installation. During this step, the installation procedure also checks to see if the command procedure SYS$UPDATE:DECW$COMPILE_ADA_UNITS.COM exists on your system. This command procedure makes the Ada DECwindows interfaces available; if the command procedure exists, the VAX Ada installation procedure reminds you that you may need to reexecute the command procedure after VAX Ada is installed.

This excerpt assumes that a Version 1.5 compiler was saved in step 5.

```
%VMSINSTAL-I-RESTORE, Restoring product saveset B...


    There will be no further questions.

    The installation will take approximately 15 more minutes.

    After the installation is complete, you need to add the following
    command to the site-specific system startup command file,
    SYS$MANAGER:SYSTARTUP_V5.COM:

       $ @SYS$STARTUP:ADA$STARTUP


                            NOTE

       The Ada startup command file is now located in
       SYS$STARTUP, as opposed to its previous location
       in SYS$MANAGER. As a result, when modifying your
       site-specific system startup command file, be sure
       to specify SYS$STARTUP. In addition, you should
       delete any existing copies of the command file
       SYS$MANAGER:ADA$STARTUP.COM after the installation
       is complete.

%ADA-I-DECWADA, After this installation is complete, you may need to
-ADA-I-DECWADA, reexecute SYS$UPDATE:DECW$COMPILE_ADA_UNITS.COM to
-ADA-I-DECWADA, obtain the Ada DECwindows interfaces.

    When the installation is complete, the previous version of the
    compiler and its environment will be in the following directory:

        SYS$COMMON:[SYSMGR.ADA$V1_5-44]

    In addition, two new command procedures will be created in
    SYS$MANAGER:
```

```
        ADA$USE_V1_5-44.COM
        ADA$USE_INSTALLED.COM

    When invoked, the first command procedure defines the logical names
    that you will need to access the previous version of the compiler.
    The second command procedure defines the logical names that you
    will need to access the currently installed version of the compiler.

%VMSINSTAL-I-SYSDIR, This product creates system directory
 VMI$ROOT:[SYSMGR.ADA$V1_5-44].
%VMSINSTAL-I-RESTORE, Restoring product saveset C ...
%ACS-I-CL_VFYOK, VMI$ROOT:[SYSLIB.ADALIB$NEW] verified
%VMSINSTAL-I-RESTORE, Restoring product saveset D ...
%VMSINSTAL-I-SYSDIR, This product creates system directory
 VMI$ROOT:[SYSLIB.ADA$SCALIB].
```

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 files created by the procedure are moved to their target directories.

Note that at this point, an informational message notifies you when the new files are moved to their target directories. However, if you are installing VAX 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.

The following files are created or modified by the installation procedure:

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$SCALIB] and contents	(new)
SYS$HELP:ADA02*n*.RELEASE_NOTES	(new)
SYS$TEST:ADA$IVP.COM	(new)

If you saved the currently existing compiler in step 5, the installation procedure also creates the following files:

SYS$COMMON:[SYSMGR.ADA$*ident_string*] and contents	(new)
SYS$MANAGER:ADA$USE_*ident_string*.COM	(new)

SYS$MANAGER:ADA$USE_INSTALLED.COM (new)

Step 8: Observe the Installation Verification Procedure.

If you chose to run the IVP in step 5, VMSINSTAL now calls the IVP to verify that VAX Ada was installed successfully.

A temporary program library is created and defined to be the current program library; the VAX Ada predefined units just installed in ADA$PREDEFINED are entered into the program library. A VAX Ada program consisting of two compilation units is compiled, linked, and run. The program makes use of the predefined package TEXT_IO to display a series of messages.

The following messages are issued during the execution of the IVP:

```
  Beginning the VAX Ada installation verification procedure.

%ACS-I-CL_LIBCRE, Library VMI$ROOT:[SYSUPD.ADA02n.ADALIB] created
%ACS-I-CL_LIBIS, Current program library is VMI$ROOT:[SYSUPD.ADA02n.ADALIB]
VAX Ada V2.n installation verification test
Testing, testing,           1,           2,           3,           4
Testing, testing
VAX Ada V2.n installation verification PASSED
```

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

Step 9: End the installation procedure.

```
   Installation of VAX Ada V2.n completed at hh:mm

   VMSINSTAL procedure done at hh:mm

$ 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, such as the VAX Ada HELP file, 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. Therefore, 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.

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; therefore, when you enter an ADA or ACS command, the newly installed compiler and program library manager are invoked. However, the VAX Ada kit is designed to allow you to have continued access to your previously existing compiler while you are using the newly installed compiler.

Note *The currently existing HELP file is not automatically saved along with the compiler. See the Installing the VAX Ada HELP File section for information on saving the currently existing HELP file.*

During the installation, you are asked (in step 5 of the Installing VAX Ada section) 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$MANAGER. The subdirectory is named ADA$*ident_string*.DIR, where *ident_string* is the ident string of the currently existing compiler. The installation procedure then copies the currently existing versions of the 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), and SCA library for the predefined units (ADA$SCALIB.DIR) to that subdirectory.

In addition, the installation procedure places two command procedures in SYS$MANAGER: ADA$USE_*ident_string*.COM and ADA$USE_INSTALLED.COM. When invoked, the first command procedure defines the appropriate logical names needed to access the previously existing compiler; the second command procedure defines the appropriate logical names to access the installed compiler.

For example, if Version 2.0 is installed as the default compiler and you want to access a saved Version 1.5-44 compiler, you would enter the following command:

```
$ @SYS$MANAGER:ADA$USE_V1_5-44.COM
```

Subsequently, when you enter an ADA or ACS command, the Version 1.5-44 compiler or program library manager is invoked.

If you want the old version of the VAX Ada compiler to be the default, modify SYS$STARTUP:ADA$STARTUP.COM by replacing the current logical definitions with the ones from SYS$MANAGER:ADA$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.

Note that 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 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 will by default use the default version of the compiler. For example, consider the situation where the Version 2.0 compiler is the currently installed compiler, but the Version 1.5 compiler has been defined to be the default compiler. If a user invokes @SYS$MANAGER:ADA$USE_INSTALLED.COM and then uses an ACS command that creates a command procedure for batch submission, the Version 1.5, not the Version 2.0, compiler will be used during the execution of the command procedure.

To override this behavior, the user can modify his or her 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 to define global VMS symbols to represent the invocations of the command procedure; for example:

```
$ V15   == "@SYS$MANAGER:ADA$USE_V1_5-44.COM"
$ V20   == "@SYS$MANAGER:ADA$USE_INSTALLED.COM"
```

Program libraries generated by a Version 2.n compiler will not be accepted by older compilers. Similarly, program libraries generated by a Version 1.n compiler cannot be used directly by a Version 2.n compiler; you must convert these libraries to Version 2.n format by using the ACS CONVERT LIBRARY command.

Note that 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 2.n commands are accepted by a Version 1.n compiler and program library manager. However, if you use a new Version 2.n qualifier or option with a Version 1.n compiler, the qualifier or option will be ignored.

If this behavior 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 2.0, but you have redefined your environment so that the Version 1.5 compiler is being used, you could enter the following command to use the Version 1.5 command definition:

```
$ SET COMMAND SYS$COMMON:[SYSMGR.ADA$V1_5-44]ADACLD.CLD
```

This technique will work only as long as you remain in the current process context. Recall that when a spawned subprocess is created (for example, you enter a DCL SPAWN, ACS COMPILE/WAIT, ACS RECOMPILE/WAIT, ACS LOAD/WAIT, or LSEDIT COMPILE command) 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.

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 merely copied or renamed to the appropriate places.

Installing VAX Ada on a VAXcluster

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

1. Enter the LICENSE LOAD command to activate the license on each node in the VAXcluster on which VAX Ada is to be executed as described in the *VMS License Management Utility Manual*.

2. If the other nodes share a common system disk with the installation node, install VAX 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 :== $INSTALL/COMMAND_MODE
   INSTALL> REPLACE SYS$LIBRARY:DCLTABLES.EXE RETURN
   ```

   These commands modify the DCL command tables on the other nodes so that the ADA and ACS commands will be 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.

4. If previous versions of the VAX Ada compiler (ADA.EXE) or program library manager (ACS.EXE) have been specified as known or shared images, VMSINSTAL automatically attempts to install the new versions as known or shared images. However, if you are running VAX Ada on a VAXcluster, VMSINSTAL automatically installs new versions as known

or shared images only on the node on which the installation is actually performed. You must manually install new versions as known or shared images on other nodes using the VMS Install Utility.

Installing the VAX Ada HELP File

This section contains excerpts from the VAX Ada HELP file installation procedure and explains each step. All of the information in the Using VMSINSTAL section applies to the HELP file installation, although not all of the VMSINSTAL command-line options are shown in the excerpts in this section. Defaults appear in brackets throughout the installation procedure.

To abort the installation procedure at any time, press CTRL/Y. When you press CTRL/Y, the installation procedure deletes all files it has created up to that point and then returns you to DCL level. If you want to retry the installation procedure after pressing CTRL/Y, you must proceed from step 2.

If you chose to keep the currently existing compiler available (in step 5 of the Installing VAX Ada section), you may also want to save the currently existing HELP file. The VAX Ada HELP file 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]V1_nADAHELP.HLP -
_$ SYS$HELP:HELPLIB.HLB
$ LIBRARY/CREATE/HELP USER:[SMITH]V1_nADAHELP.HLB -
_$ USER:[SMITH]V1_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]V1_nADAHELP.HLB
```

Step 1: Log in to a privileged account and set your default device and directory to SYS$UPDATE.

```
RETURN
Username: SYSTEM   RETURN
Password:          RETURN
$ SET DEFAULT SYS$UPDATE  RETURN
```

Step 2: Invoke VMSINSTAL.

```
$ @VMSINSTAL ADAHLP02n MUA0: RETURN

        VAX/VMS Software Product Installation Procedure VX.X


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

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

If you are satisfied with the backup of your system disk, press RETURN. Otherwise, type NO and press RETURN to discontinue the installation. Digital recommends that you back up your system disk before installing any software.

Step 3: Insert the installation kit volume.

```
Please mount the first volume of the set on MUA0:.
* Are you ready? Y RET
The following products will be processed:

  ADAHLP V2.n


        Beginning installation of ADAHLP V2.n at hh:mm


%VMSINSTAL-I-RESTORE, Restoring product saveset A...
%VMSINSTAL-I-RELMOVED , The product's release notes have been
  successfully moved to SYS$HELP.
```

To indicate that you want to continue the installation, mount volume 1, type YES, and press RETURN.

Step 4: Read informational messages.

The VAX Ada HELP file installation procedure produces a number of informational messages that report on the progress of the installation.

```
   There will be no further questions. The installation
   will take approximately 2 more minutes.

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

If the installation procedure is successful, the SYS$HELP:HELPLIB.HLB file is modified to include the new VAX Ada HELP library. Also, the file SYS$HELP:ADAHLP02*n*.RELEASE_NOTES is created.

Step 5: End the installation procedure.

```
        Installation of ADAHLP V2.n completed at hh:mm


        VMSINSTAL procedure done at hh:mm

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

When the VAX Ada HELP file 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.

After the Installation

VAX Ada is now installed and the VAX Ada compiler can be invoked by all users with the ADA command; the VAX 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. All 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_V5.COM file, as requested during step 7 in the Installing VAX Ada section:

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

If VAX Ada will be used extensively on your system, you can reduce system overhead and memory requirements by installing the compiler, program library manager, and run-time library as shared images using the Install Utility. For more information, see the Installing VAX Ada as a Shared Image section.

To ensure device independence when backing up and restoring VAX 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 *VMS DCL Dictionary* and the *Guide to VMS File Applications*. For information on using concealed device logical names with VAX Ada program libraries, see *Developing Ada Programs on VMS Systems*.

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 if system problems occur, execute the following command procedure:

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

See step 8 of the Installing VAX Ada section for more information on what happens during the IVP.

Error Conditions

If the VAX Ada installation procedure, VAX Ada HELP file installation procedure, or VAX Ada IVP fails for any reason when you are running VMSINSTAL, a message like the following is displayed:

```
%VMSINSTAL-E-INSFAIL, The installation of VAX Ada V2.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 the Installation Procedure Requirements section).
- Process quotas required by VMSINSTAL are incorrect (see the Using VMSINSTAL section).
- The VMS 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 <REFERENCE>(VMS_MESSAGES_REF), your processor-specific installation/operations guide, or the *VMS 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 increase an authorized quota value.

If the installation fails, you must restart the installation procedure from step 2. If the installation fails due to an IVP failure and the product's key has been registered and loaded, contact a Digital field service representative.

Installing VAX Ada as a Shared Image

If VAX Ada will be used extensively on your system, you can reduce the system overhead and memory requirements by installing it as a shared image. (The VAX Ada run-time library is bundled with the VMS operating system, but, like the compiler and program library manager, it is not automatically installed as a shared image unless it was previously installed as a shared image.) To perform the installation on a system that is currently running, use the VMS Install Utility while you are logged in to a privileged account. If errors occur, see <REFERENCE>(VMS_MESSAGES_REF).

First, determine the number of available global pages and global sections on your system. See the Installation Procedure Requirements section for information on how to do this.

Then, invoke the VMS Install Utility and install VAX Ada as a shared image by typing the following commands:

```
$ RUN SYS$SYSTEM:INSTALL RETURN

INSTALL> REPLACE SYS$SYSTEM:ADA.EXE/OPEN/SHARED/HEADER_RESIDENT RETURN
INSTALL> REPLACE SYS$SYSTEM:ACS.EXE/OPEN/SHARED/HEADER_RESIDENT RETURN
INSTALL> EXIT RETURN
```

By using the /HEADER_RESIDENT qualifier, you save disk I/O on image activation (at the expense of a permanently allocated page of memory). By using the REPLACE command, you guarantee that the installation will work whether the images are newly installed or were previously installed. If a file has not been previously installed, you may get a warning message; however, you can ignore the message because the file will be installed correctly.

To ensure that VAX Ada is installed as a shared image each time the system is bootstrapped, include the following lines in SYS$MANAGER:SYSTARTUP_V5.COM (the site-specific startup file):

```
$ INSTALL :== $INSTALL/COMMAND_MODE
$ INSTALL
REPLACE SYS$SYSTEM:ADA.EXE/OPEN/SHARED/HEADER_RESIDENT
REPLACE SYS$SYSTEM:ACS.EXE/OPEN/SHARED/HEADER_RESIDENT
EXIT
```

Maintenance Updates

Digital may periodically issue maintenance updates of VAX 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 a maintenance update is released, the version number changes. For example, if the current version is 2.0, the version number of the first maintenance update will be 2.1. In addition, each maintenance update includes new release notes. The release notes describe the changes that have been made to VAX Ada since the previous release.

The update release notes are provided on line. You should read the release notes when you first install VAX 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 VAX Ada, follow the installation procedure through step 4. To locate the release notes after VAX Ada is installed, display or print the file SYS$HELP:ADA02*n*.RELEASE_NOTES (where *n* is the number of the current maintenance update), or type the following command:

```
$ HELP Ada Release_Notes RETURN
```

To locate the release notes after the VAX Ada HELP file is installed, display or print the file SYS$HELP:ADAHLP02*n*.RELEASE_NOTES (where *n* is the number of the current maintenance update).

Improving Product Performance

This section gives detailed information on how to improve the performance of the VAX Ada compiler and program library manager.

Memory Usage

Working set size and virtual address space are important factors in achieving efficient compilation of VAX Ada programs. The following sections analyze the results of sample compilations and suggest general guidelines for adjusting working set size and selecting an appropriate virtual address space.

Working Sets To measure the effects of working set size on page faults and compilation rates, selected VAX Ada programs representing typical usage were compiled with a variety of working set sizes. The test compilations were run on a standalone system with the following configuration: a 13-megabyte VAXstation II with an RD53 system disk drive and an RD53 user disk drive running VMS Version 5.1.

Table 1 gives brief descriptions of the source files compiled in the tests (each file represents one or more VAX Ada program units).

Table 1 Description of Test Programs

Programs	Descriptions
BL6PERF0	250 lines (27 disk blocks). A set of eight library package specifications that were taken from the front end of a compiler written in Ada. The set includes units for lexical analysis and parsing.
BL6PERF1	850 lines (56 disk blocks). One of the package bodies that goes with BL6PERF0. The **with** clause names seven packages (five from BL6PERF0). The body includes one instantiation of INTEGER_IO.TEXT_IO.
ADABNF	3400 lines (284 disk blocks). A library package specification that provides the control tables for the LALR parser in BL6PERF0. This package specification consists of a small number of array objects initialized by very large aggregates. One of these is an array aggregate consisting of 2700 named associations in which each component value is itself a record aggregate with two components. Another is similar, with nearly 500 associations in which each component value is a record aggregate with three components.
ADAMACROS	125 lines (10 disk blocks). A complete program (library procedure) that performs a simple macro-like substitution for a set of files using a simple definition file. (Used to construct the VAX Ada-specific versions for the implementation-dependent tests of the Ada validation tests.)
STARLET0	29,000 lines (2113 disk blocks). Two library package specifications consisting of CONDITION_HANDLING and STARLET as found in the predefined library for VAX Ada.
STARLET1	6 lines (1 disk block). A very small procedure that uses a single named number from package STARLET.
NULLP	4 lines (1 disk block). A procedure consisting of only a null statement.

Each of these units was compiled using working sets of 500, 750, 1000, 1250, 1500, 1750, 2000, 2500, 3000, 4000, 6000, and 10000 pages.

For these experiments, the working set size was established using the following DCL command, where x is the desired working set size in pages:

```
$ SET WORKING_SET/LIMIT=x/QUOTA=x/NOADJUST
```

This causes the working set to be fixed as specified and disables the VMS automatic adjustment strategies.

Effect of Working Set on Paging Rate Page fault data is shown in Figure 1. The vertical axis is the number of page faults times 1000. For the program STARLET0, the number of page faults is scaled by a multiplier of 10.

The significant feature of the graph in Figure 1 is the working set associated with the knee of each curve. As shown, the knee occurs at around 2500 pages (1-1/4 megabytes) in each case.

The value of 2500 pages is significant because it indicates that as the working set is reduced below this value, the amount of paging rises rapidly. This increased paging translates into increased system load that affects the overall system performance and responsiveness for all users. On the other hand, as the working set is increased above this figure, the paging does not decrease very rapidly. Thus, a working set of 2500 pages should be considered as the minimum required for efficient VAX Ada compilations. (This figure is also reinforced by the analysis of compilation rates.)

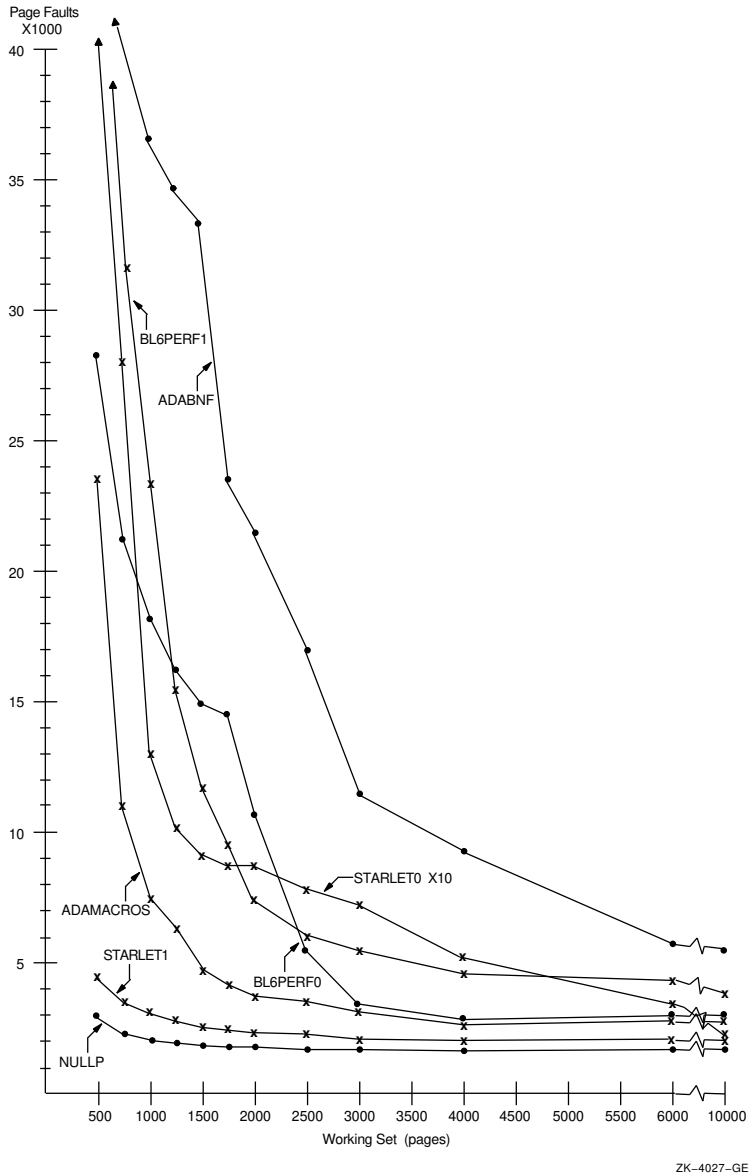
Effect of Working Set on Compilation Rate Using data derived from the sample compilations used to analyze page faulting behavior, Figure 2 shows the compilation rate (measured in source lines per elapsed minute) in relation to working set size. These compilation rates are based on the number of source lines compiled per elapsed minute and not on the number of source lines compiled per CPU minute, because the number of source lines per elapsed minute is a better measure of compilation rate throughput. The vertical axis in Figure 2 is the number of lines compiled per elapsed minute, which is scaled by a multiplier shown on the graph for each program.

In Figure 2, there is a clearly defined knee at 2500 pages, for which the following determinations can be made:

- At working sets above 2500 pages, there is very little increase in the compilation rate.
- At working sets below 2500 pages, there is a decrease in the compilation rate due to the paging overhead.

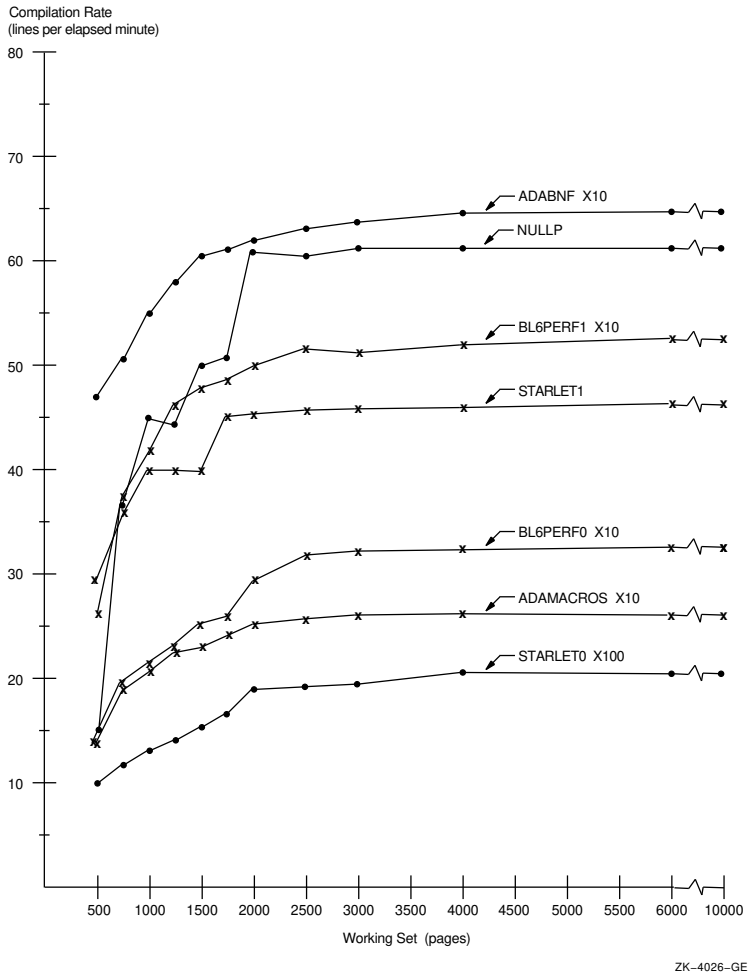
These measurements were made under standalone conditions on a system with a relatively large amount of memory, so that even at very small working set sizes, page faults could be quickly and cheaply satisfied from VMS cached pages in main memory. On a more heavily loaded system, or one with less main memory, the cost of paging is likely to be higher as more page faults require disk access. This increase in page faults would result in more sharply reduced compilation rates and throughput at the smaller working sets.

Figure 1 Page Faults Versus Working Set Size



It may seem surprising that such a relatively large working set is desirable even for very small programs. Part of the explanation for this is the rather large size of the VAX Ada compiler itself—approximately 4600 disk blocks. The operation of just paging in the VAX Ada compiler during its execution phase

Figure 2 Compilation Rate Versus Working Set Size



causes more paging than other compilers. (Other VMS compilers supplied by Digital are generally in the 600- to 1200-block range.)

Suggestions for Controlling Working Set Sizes Some VMS systems are used in environments where a large number of users coexist with relatively small working set quotas (an educational environment is a key example). In such settings, VAX Ada may appear both to be very sluggish for each individual user and to induce a large system overhead affecting all users. A suggested practice is to establish a special batch job queue intended primarily for VAX Ada compilations. (While especially important in such cases, this practice is a good one even on systems where there is generous memory for large user quotas.)

The characteristics of the queue should be set to allow a fairly large working set quota (at least 2500 pages) with a working-set extent of 4000 or more for each of the concurrent compilations operating at normal interactive priority (priority 4). In general, you can allow as many concurrent compilations as you have VAX units of performance (VUPs). (One VUP is equal to the performance of a VAX-11/780.)

These characteristics will allow VAX Ada compilations to complete efficiently and without inducing excessive system overhead. You may need to experiment a bit with the various parameters and options to find the configuration that works best on your system.

The ACS COMPILE and RECOMPILE commands are designed to support this style of batch processing. By default, the COMPILE and RECOMPILE commands submit compilations to the queue named by the logical name ADA$BATCH. If ADA$BATCH is not defined, then the system batch queue SYS$BATCH is used. When a separate queue for VAX Ada compilations is desired, ADA$BATCH should be defined as a system logical name whose translation is the name of the appropriate queue.

Virtual Address Space It is useful to consider the peak virtual memory required for the compilation of each unit. This is the total virtual memory required for the execution of the process for all purposes, including the compiler itself, as well as dynamic memory used for representing the program, input-output buffers, and so on.

The virtual memory requirements of the programs compiled for the experiments on working set size are as follows:

Program	Peak Virtual Pages
BL6PERF0	9100
BL6PERF1	10300
ADABNF	12000
ADAMACROS	9100
STARLET0	18900
STARLET1	8900
NULLP	8900

These figures suggest that in a large-scale production environment the system and individual quotas should be configured to allow a virtual address space of at least 20,000 pages (10 megabytes) for VAX Ada compilations. In less demanding environments, 15,000 pages might generally be adequate. (The actual virtual memory used and remaining for a compilation is displayed as part of the summary in the compilation listing.)

See the Resource Requirements section for more information on specific system parameters that affect the amount of available virtual address space.

Resource Requirements

The following sections provide information on the system resources used by the VAX Ada compiler and program library manager. Key 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 VAX Ada requires larger values than the VMS defaults, or for which the values are larger than those required for other VAX languages.

The VMS operating system controls resource usage through two primary sets of parameters: system-wide parameters and per-process parameters. You use the SYSGEN utility to define and modify system-wide 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 VMS resources, SYSGEN, and UAF parameters is contained in the *Guide to Setting Up a VMS System*. 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. See the *Guide to Setting Up a VMS System* for information on how to use AUTOGEN and how to specify parameter values before invoking AUTOGEN.

ASTLM—AST Queue Limit Parameter The UAF AST queue limit (ASTLM) parameter limits the sum of thefollowing:

- 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 VAX Ada delay statement is implemented as a call to the VMS SYS$SETIMR system service, which executes an AST routine. Also, VAX Ada provides facilities for calling VMS system routines that execute AST routines; for example, the package STARLET provides the system services SYS$QIO and SYS$QIOW, both of which allow you to 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 VMS typical value for ASTLM is 24; this should be sufficient for most Ada programs. However, if your programs involve tasks that could execute many delay statements simultaneously, or you use a high number of calls to VMS system routines that execute AST routines, you may want to increase this value.

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

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

VAX Ada and the VAX Ada program library manager use VAX RMS to access the VAX Ada program library index file, ADALIB.ALB. Up to eight locks may be used for each library or sublibrary that is open. When a sublibrary is opened, it is usually the case that its parent and its ancestors are also opened. Therefore, 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 may have two libraries or sublibraries open at the same time, which doubles the number of locks.

The suggested VMS typical value for ENQLM is 30. This value is not sufficient for using the VAX Ada program library manager and sublibraries; the recommended value for VAX Ada users is 60.

FILLM—Open File Limit Parameter 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. (See the Program Library Networking Effects section for more information about logical links.)

The VAX Ada compiler and program library manager 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 FILLM is set to 20, the compilation will be slower than if FILLM had been set to 100.

FILLM is a pooled limit with a suggested typical value of 20. This value is not sufficient for most VAX Ada compilations; the recommended minimum value for VAX Ada users is 50. Note, however, that the value of FILLM must always be lower than the value of the SYSGEN channel count (CHANNELCNT) parameter. See the Channel Count Parameters section for more information.

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.

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

The VAX Ada program library manager creates a subprocess to run the linker or the VAX 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 VMS typical value for PRCLM is 2; this should be sufficient for the VAX Ada program library manager as well.

TQELM—Timer Queue Entry Limit Parameter The 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 VAX Ada delay statement is implemented as a call to the VMS SYS$SETIMR system service, which adds an entry to the timer queue.

The suggested VMS typical value for TQELM is 20; this value should be sufficient for most Ada programs. However, if your programs involve tasks that could execute many delay statements simultaneously, you may want to increase this value.

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

Virtual Memory Usage The Memory Usage section describes the VAX Ada compiler's requirements for virtual address space and working sets. Various SYSGEN and user parameters should be set so that a VAX Ada compilation can use up to 20,000 pages (10 megabytes) of virtual memory. Working set parameters (see the following sections) should be adjusted to provide good performance for such large virtual address spaces.

VIRTUALPAGECNT—Maximum Number of Virtual Pages Parameter The SYSGEN VIRTUALPAGECNT parameter sets the maximum number of virtual pages that can be mapped for any one process. The VMS default (8192 pages, or 4 megabytes) is too small for the compilation of large VAX Ada programs; a value of 20,000 (10 megabytes) is recommended.

You should also make sure that the system paging file is large enough to accommodate processes with large page file quotas. A paging file size of 30,000 blocks is adequate for single users when the value of VIRTUALPAGECNT is 20,000.

PGFLQUOTA—Paging File Quota Parameter The UAF paging file quota (PGFLQUOTA) parameter limits the number of pages that your process can use in the system paging file. In effect, 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 VMS typical value of 12800 for PGFLQUOTA is too small; a value of 17000 (VIRTUALPAGECNT—3000) is recommended.

You should also make sure that the system page file is large enough to accommodate processes with large page file quotas. The size of the paging file should be greater than the maximum paging file quota for an individual. If your value of PGFLQUOTA or the system's value of VIRTUALPAGECNT is too small, large Ada compilations will fail with this 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 VAX 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 larger than the virtual page count (SYSGEN VIRTUALPAGECNT parameter). For example, if the value of VIRTUALPAGECNT is 20,000 pages, then a paging file of at least 30,000 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 will appear on the operator's console:

```
%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 operations to adjust the size of the system paging file after adjusting the values of VIRTUALPAGECNT and PGFLQUOTA. Note that AUTOGEN sometimes chooses a paging file size that is too low; you may need to explicitly specify a paging file size before invoking the AUTOGEN operation. See the

Guide to Setting Up a VMS System for information on how to use AUTOGEN and how to specify parameter values before invoking AUTOGEN.

See the VIRTUALPAGECNT—Maximum Number of Virtual Pages Parameter and PGFLQUOTA—Paging File Quota Parameter sections for more Ada-related information on these parameters. See the *Guide to Setting Up a VMS System* for information on the VMS default value of PGFLQUOTA; see the *VMS System Generation Utility Manual* for more information on the VMS default value of VIRTUALPAGECNT.

WSQUOTA and WSEXTENT—Working Set Quota and Extent 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. In other words, the system guarantees the user that WSQUOTA physical pages will be 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 VMS operating system willattempt to provide additional physical pages to a process that is page-faulting heavily. Thus, on a lightly loaded system, the user's working set can grow beyond WSQUOTA up to WSEXTENT.

The discussion in the Memory Usage section shows that a VAX Ada compilation should have a working set of 2500 pages or more. If a user is going to do VAX Ada compilations interactively or in a subprocess (COMPILE/WAIT), the user's UAF entry should specify WSQUOTA=2500.

Many sites may not permit such large working sets for interactive jobs. In this case, a batch queue should be established for VAX Ada compilations. The VAX 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.

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

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

A typical VAX 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
```

WSMAX—Working Set Maximum Number of Pages Parameter The SYSGEN working set maximum number of pages (WSMAX) parameter sets the maximum number of pages for any working set on a system-wide basis. The value of WSMAX should be as large as the maximum WSEXTENT value assigned to any user or batch queue. The VMS default WSMAX of 1024 pages is too small for VAX Ada; a value of at least 4000 is recommended.

Program Library Networking Effects There are several SYSGEN and DECnet parameters (on both the local and remote nodes) that may 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 may 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 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 will also reduce the potential number of logical links created to the remote node: a logical link is only required for files that are not accessed locally.

Also note that the creation of a logical link may also involve the creation of a process on the remote node. Thus, you may need to increase (or at least monitor) the values of the SYSGEN parameters MAXPROCESSCNT and BALSETCNT to allow 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 *Guide to DECnet-VAX Networking*.

After you have set or reset system parameters to accommodate the use of remote nodes, you may want to run the Digital-supplied AUTOGEN command procedure (SYS$UPDATE:AUTOGEN.COM) to recompute optimal values for related parameters.

Channel Count Parameters The SYSGEN channel count (CHANNELCNT) parameter specifies the maximum number of channels for each process in a VMS 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 VAX Ada compiler and program library manager 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 compilation 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 Ada compilations that involve a large number of compilation units; however, you should be careful not to raise the value of FILLM above the value of CHANNELCNT.

Note that the value of the FILLM parameter may interact with other system parameters; see the FILLM—Open File Limit Parameter and Program Library Networking Effects sections for more information on the FILLM parameter.

Determining and Reporting Problems

If an error occurs while you are using VAX Ada and you believe that the error is caused by a problem with VAX Ada, take one of the following actions:

- If you purchased VAX Ada within the past 90 days and you think the problem is caused by a software error, you can submit a Software Performance Report (SPR).
- If you have a Basic or DECsupport Software Agreement, you should call your Customer Support Center. With these services, you receive telephone support that provides high-level advisory and remedial assistance. For more information, contact your local Digital representative.
- If you have a Self-Maintenance Software Agreement, you can submit a Software Performance Report (SPR).

If you find an error in the VAX Ada documentation, you should fill out and submit a Reader's Comments form from the back of the document in which the error was found. Include the section and page number where the error was found.

When you prepare to submit an SPR, please take the following steps:

1. Describe as accurately as possible the state of the system and the circumstance when the problem occurred. Include in the description the version number of VAX 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 (floppy diskette or magnetic tape). If necessary, also submit a copy of the program library used to build the application. Use the VMS Backup Utility to copy the program library to the machine-readable media. All media will be returned to you when the SPR is answered.
6. Report only one problem per SPR. This will facilitate a more rapid response.
7. Mail the SPR package to Digital.

Experience shows that many SPRs 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.

Sample Installation

This section contains a sample installation of VAX Ada. Note that 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 VAX Ada* letter before you install this product. This letter notes any significant changes in the installation that vary from this sample.

```
$ @VMSINSTAL ADA020 MUA0: OPTIONS N

        VAX/VMS Software Product Installation Procedure V5.0

It is 25-APR-1989 at 19:11.
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 V2.0

        Beginning installation of ADA V2.0 at 19:11
```

```
%VMSINSTAL-I-RESTORE, Restoring product saveset A ...

    Release Notes Options:

        1.  Display release notes
        2.  Print release notes
        3.  Both 1 and 2
        4.  Copy release notes to SYS$HELP
        5.  Do not display, print or copy release notes

* Select option [2]: 4 RETURN
* Do you want to continue the installation? YES RETURN
%VMSINSTAL-I-RELMOVED , The product's release notes have been
 successfully moved to SYS$HELP.
* Do you want to run the IVP after the installation [YES]? RETURN
* Do you want to include the SCA library for the predefined
  units [YES]? ? RETURN

This version provides support for the VAX Source Code Analyzer (SCA).  If
you do not use SCA, you may save about 10,000 blocks by not installing the
SCA library for the predefined units.  If you choose to install the SCA
library, the sources for the predefined units (about 7,000 additional
blocks) are automatically installed.

* Do you want to include the SCA library for the predefined
  units [YES]? RETURN
* Do you want to save the currently existing compiler [YES]? ? RETURN

Support has been added to allow you to save the currently existing compiler
and its environment(ACS, ADARTL, ADAMSG, ADA$PREDEFINED,ADA$SCA_PREDEFINED).
By saving the existing compiler, you can have continued access to either the
existing compiler or the compiler that you are installing.

Note the existing compiler will not be saved if it is the same version
as the one you are installing.

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

        Product:      ADA
        Producer:     DEC
        Version:      2.0
        Release Date: 15-MAR-1989


* Does this product have an authorization key registered and
  loaded? YES RETURN
%VMSINSTAL-I-RESTORE, Restoring product saveset B ...

    There will be no further questions.

    The installation will take approximately 15 more minutes.

    After the installation is complete, you need to add the following
    command to the site-specific system startup command file,
    SYS$MANAGER:SYSTARTUP_V5.COM:

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

```
                                NOTE

        The Ada startup command file is now located in
        SYS$STARTUP, as opposed to its previous location
        in SYS$MANAGER.  As a result, when modifying your
        site-specific system startup command file, be sure
        to specify SYS$STARTUP.  In addition, you should
        delete any existing copies of the command file
        SYS$MANAGER:ADA$STARTUP.COM after the installation
        is complete.

%ADA-I-DECWADA, After this installation is complete, you may need to
-ADA-I-DECWADA, reexecute SYS$UPDATE:DECW$COMPILE_ADA_UNITS.COM to
-ADA-I-DECWADA, obtain the Ada DECwindows interfaces.

    When the installation is complete, the previous version of the
    compiler and its environment will be in the following directory:

        SYS$COMMON:[SYSMGR.ADA$V1_5-44]

    In addition, two new command procedures will be created in
    SYS$MANAGER:

        ADA$USE_V1_5-44.COM
        ADA$USE_INSTALLED.COM

    When invoked, the first command procedure defines the logical names
    that you will need to access the previous version of the compiler.
    The second command procedure defines the logical names that you
    will need to access the currently installed version of the compiler.

%VMSINSTAL-I-SYSDIR, This product creates system disk directory
 VMI$ROOT:[SYSMGR.ADA$V1_5-44].
%CREATE-I-EXISTS, VMI$ROOT:[SYSMGR.ADA$V1_5-44] already exists
%VMSINSTAL-I-RESTORE, Restoring product saveset C ...
%ACS-I-CL_VFYOK, VMI$ROOT:[SYSLIB.ADALIB$NEW] verified
%VMSINSTAL-I-RESTORE, Restoring product saveset D ...
%VMSINSTAL-I-SYSDIR, This product creates system disk directory
 VMI$ROOT:[SYSLIB.ADA$SCALIB].
%VMSINSTAL-I-MOVEFILES, Files will now be moved to their target
 directories...

  Beginning the VAX Ada installation verification procedure.

%ACS-I-CL_LIBCRE, Library VMI$ROOT:[SYSUPD.ADA020.ADALIB] created
%ACS-I-CL_LIBIS, Current program library is VMI$ROOT:[SYSUPD.ADA020.ADALIB]
VAX Ada V2.0 installation verification test
Testing, testing,           1,           2,           3,           4
Testing, testing
VAX Ada V2.0 installation verification PASSED
        Installation of ADA V2.0 completed at 19:35

        VMSINSTAL procedure done at 19:36

$ LOGOUT
  SYSTEM       logged out at 25-APR-1989 19:43:43.00
```