DEC C/C++ Run-Time Components for OpenVMS VAX

Reference and Installation Guide

Order Number: AA–Q0XBA–TK

October 1993

This guide contains instructions for installing the DEC C/C++ Run-Time Components on OpenVMS VAX systems. It also contains important information on distributing applications that are linked to DEC C/C++ Run-Time Components. This guide applies to DEC C/C++ Run-Time Components Version 6.0 for OpenVMS VAX and all maintenance updates throughout that version.

Revision/Update Information:	This is a new guide.
Operating System & Version:	OpenVMS VAX Version 5.5 or higher. Future maintenance releases may require higher versions.
Software Version:	DEC C/C++ Run-Time Components Version 6.0 for OpenVMS VAX

Digital Equipment Corporation
Maynard, Massachusetts

First Printing, October 1993

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

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

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

Restricted Rights: Use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227-7013.

© Digital Equipment Corporation 1993.

All Rights Reserved.

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

The following are trademarks of Digital Equipment Corporation: AXP, DEC, DECmcc, Digital, DECnet, DECthreads, DECwindows, OpenVMS, VAX, VAX C, VAX DOCUMENT, VMS, VMScluster, and the DIGITAL logo.

The following is a third-party trademark:

PostScript is a registered trademark of Adobe Systems Inc.

ZK6363

This document is available on CD-ROM.

This document was prepared using VAX DOCUMENT, Version 2.1.

Contents

Preface v

1 Distributing Dependent Products

1.1 Impact on Software Development and Distribution 1–1
1.1.1 Linking Against Older Versions 1–3
1.1.2 Redistributing the DEC C/C++ Run-Time Components Kit 1–3
1.1.3 Linking with Object Libraries 1–4
1.2 Saving the Old System Development Environment 1–4
1.3 Notifying Site Users 1–6
1.4 Diagnosing the Distribution Problem 1–6
1.5 Symptoms and Appropriate Responses 1–8
1.6 Requirements for Redistribution of the DEC C/C++ Run-Time Components Kit 1–9
1.7 Saving the DEC C/C++ Run-Time Components Kit to Allow Redistribution 1–9

2 Preparing for Installation

2.1 Required Operating System Components and Software 2–1
2.2 Installation Procedure Requirements 2–2
2.2.1 Process Privileges 2–2
2.2.2 Process Account Quotas 2–2
2.2.3 System Parameters 2–3
2.2.4 Disk Space 2–4
2.2.5 Backing Up Your System Disk 2–5
2.2.6 Additional System Maintenance Procedures 2–5
2.2.6.1 Checking System Parameter Values 2–5
2.2.6.2 Calculating the Values for Global Pages and Global Sections 2–5
2.2.6.3 Changing System Parameter Values with AUTOGEN . . . 2–6
2.2.6.4 Setting Dynamic System Parameter Values 2–7

3 Installing the DEC C/C++ Run-Time Components

3.1 Installation Procedure 3–1
3.2 Error Recovery 3–10

4 After Installation

4.1 Running the Installation Verification Procedure Separately 4–1
4.2 Editing the System Files 4–2
4.3 VMScluster Considerations 4–2
4.4 Reinstalling After an OpenVMS VAX Intermediate Upgrade 4–3
4.5 Reporting Problems 4–3

A Sample Installation

Examples

A–1 Sample DEC C/C++ Run-Time Components Installation on an OpenVMS VAX System A–2

Tables

1 Conventions Used in this Guide vi
1–1 New Image Names Not Previously Available 1–7
1–2 New Image Names and Identifiers Replacing Existing OpenVMS Versions 1–8
2–1 Process Quotas for the Installing Account 2–2
2–2 Minimum Required System Parameter Values 2–3
2–3 Disk Space Requirements 2–4

Preface

This guide describes how to install the DEC C/C++ Run-Time Components on VAX processors running the OpenVMS operating system. It is intended for system managers.

Keep this guide with your distribution kit. You will need it to install maintenance updates or to reinstall the DEC C/C++ Run-Time Components kit for any other reason.

If you have any comments or suggestions concerning this guide, you may send them by electronic mail to the following Internet address:

cxx_docs@casdoc.enet.dec.com

If you have access to a FAX machine, you can also send comments or suggestions to:

DEC C++ Documentation
603-881-0120

Document Structure

This guide contains four chapters and one appendix:

- Chapter 1 describes the preparations for supporting remote software development and distribution activities.
- Chapter 2 describes how to prepare for DEC C/C++ Run-Time Components Version 6.0 for OpenVMS VAX installation.
- Chapter 3 describes how to install the DEC C/C++ Run-Time Components Version 6.0 for OpenVMS VAX.
- Chapter 4 describes postinstallation procedures.
- Appendix A shows a sample installation.

Associated Documentation

For information on the DEC C/C++ Run-Time Components Version 6.0 for OpenVMS VAX online release notes, see Chapter 2.

For general information on OpenVMS VAX system management, see the OpenVMS operating system documentation set.

For help understanding OpenVMS VAX system error messages, see the *OpenVMS System Messages and Recovery Procedures Reference Manual* or the online HELP/MESSAGE facility.

For additional information about DECthreads, see the *Guide to DECthreads*.

Conventions Used in This Guide

Table 1 lists the conventions used in this guide.

Table 1 Conventions Used in this Guide

Convention	Meaning
UPPERCASE TEXT	Uppercase text indicates the name of a command, a file, a parameter, a procedure, or utility.
$ **@SYS$UPDATE:VMSINSTAL**	In interactive examples, prompts or displayed text appears in a monospace font. User input appears in bold monospace font.
Ctrl/*x*	In interactive examples, a sequence such as Ctrl/*x* indicates that you must hold down the key labeled Ctrl while you press another key or a pointing device button; for example, Ctrl/C or Ctrl/Z.
[YES]	Square brackets indicate that the enclosed item is a default value in a VMSINSTAL prompt. Also, square brackets serve as delimiters for a directory name in a file specification.

1
Distributing Dependent Products

This chapter discusses the preparations sites must make if they support software development or distribution activities.

1.1 Impact on Software Development and Distribution

The DEC C/C++ Run-Time Components kit contains shareable image run-time components. When the DEC C/C++ Run-Time Components kit is installed on OpenVMS VAX Version 5.5, 5.5-1, or 5.5-2 systems, some of the shareable image run-time components replace older versions of existing OpenVMS components. This makes the installation of the DEC C/C++ Run-Time Components kit similar to an OpenVMS upgrade for sites that develop or distribute software for OpenVMS VAX Version 5.5, 5.5-1, or 5.5-2 systems. Software linked against any of the run-time images provided by the DEC C/C++ Run-Time Components kit cannot be executed at sites having older versions of the files in their SYS$LIBRARY directories or at sites that do not have the new images available.

Specifically, when executing the image resulting from a software build procedure that links an application against any of the new run-time components provided on this kit (see Table 1–1 and Table 1–2), a user at the installing site will encounter errors if that user subsequently copies the resulting image file to another site (target site) that is running OpenVMS VAX Version 5.5, 5.5-1, or 5.5-2 and does not have the new versions of the run-time components in their SYS$LIBRARY.

For example, consider a software application, VAX APP-XYZ, which has been developed on a cluster named GRAPES::, that is running OpenVMS VAX Version 5.5. As part of the build of VAX APP-XYZ, an executable image named XYZ.EXE is linked against the VAX C Run-Time Library (RTL) executable (SYS$LIBRARY:VAXCRTL.EXE). VAX APP-XYZ has been built and distributed to other OpenVMS VAX sites that have a version of SYS$SYSTEM:VAXCRTL.EXE at least as recent as the one distributed with OpenVMS Version 5.5 and executes properly.

However, if the system manager for the GRAPES:: cluster installs the DEC C/C++ Run-Time Components kit, one component of the kit is a new version of VAXCRTL.EXE (a more recent version than the version that exists in OpenVMS VAX Version 5.5-2). Installing the DEC C/C++ Run-Time Components kit places this newer version in SYS$LIBRARY. Subsequently, if the VAX APP-XYZ software developers execute their build procedure, the resulting version of XYZ.EXE will now depend on the new version of VAXCRTL.EXE rather than the old version. If an offsite client takes delivery of the new version of VAX APP-XYZ, users at the remote site will encounter image activation failure while trying to execute XYZ.EXE. The remote user's system issues the following message indicating that the necessary version of SYS$LIBRARY:VAXCRTL.EXE is absent:

```
$ run xyz
%DCL-W-ACTIMAGE, error activating image VAXCRTL
-CLI-E-IMGNAME, image file SYS$LIBRARY:VAXCRTL.EXE;1
-SYSTEM-F-SHRIDMISMAT, ident mismatch with shareable image
```

This problem can arise when new components such as DECC$SHR.EXE (the DEC C Run-Time Library) or CXXL$011_SHR.EXE (the DEC C++ Class Library) are linked against. It can also occur when existing components like VAXCRTL.EXE (as in this example) are used that have been replaced by newer versions from the DEC C/C++ Run-Time Components kit. Because the run-time components contained on the DEC C/C++ Run-Time Components kit are in common use, such problems can arise even for programs that do not use the DEC C or DEC C++ compilers.

Warning

Any application that links directly or indirectly against a version of VAXCRTL.EXE, VAXCRTLG.EXE, CMA$RTL.EXE, CMA$OPEN_RTL.EXE, CMA$LIB_SHR.EXE, or CMA$OPEN_LIB_SHR.EXE is affected by this problem. A linkage is indirect if a shareable image that references any of these images is linked into the application.

For simplicity, the rest of this chapter refers to ''DEC C/C++ Run-Time Components software distribution problems'' as ''distribution problems.''

For software developers who want to redistribute software that is developed using systems where the DEC C/C++ Run-Time Components kit is installed, the options are:

- Link against older versions of the shareable images
- Redistribute the DEC C/C++ Run-Time Components kit

- Link with object libraries rather than shareable images

The following sections describe these options.

1.1.1 Linking Against Older Versions

Sites that have existing build procedures (that support development and distribution of software linked against the installed version of OpenVMS VAX and want to continue to link against shareable images) must preserve the existing link environment to eliminate the distribution problem without having to modify application build procedures. This requires some preparation and notification on the part of the DEC C/C++ Run-Time Components installer.

Before installing the DEC C/C++ Run-Time Components kit, make a copy of the files in the old SYS$LIBRARY directory. After installing DEC C/C++ Run-Time Components but before executing the standard link operations, applications that need to link with pre-DEC C/C++ Run-Time Components versions of the shareable images can be supported without change by temporarily defining the appropriate logical name (either SYS$LIBRARY or SYS$SHARE) to point to the old SYS$LIBRARY. With this method, the end user of the application sees no change from the previous, unmodified use of SYS$LIBRARY. The temporary definition of SYS$LIBRARY is not needed at run time.

Whether or not the end-user site has the DEC C/C++ Run-Time Components installed does not matter if this method is used to build the application. See Section 1.2 for more detailed instructions on preparing for and using this method.

This method does not work for new components (such as DECC$SHR.EXE or CXXL$011_SHR.EXE) not previously supplied on OpenVMS VAX Version 5.5, 5.5-1, or 5.5-2 systems. You can only use these new components with newly developed applications. Applications that encounter the distribution problem from the use of new components on the DEC C/C++ Run-Time Components kit can only solve the distribution problem by redistributing the DEC C/C++ Run-Time Components kit itself. See Section 1.6 for the requirements and setup needed for redistribution of the DEC C/C++ Run-Time Components kit.

1.1.2 Redistributing the DEC C/C++ Run-Time Components Kit

Redistributing the DEC C/C++ Run-Time Components kit requires minimal changes to the development system and procedures. It does not sacrifice performance or degrade system resources. It does require that the DEC C/C++ Run-Time Components kit and this guide be shipped along with the application that requires the use of any of the components contained in it.

When new applications involving new components of the DEC C/C++ Run-Time Components kit, or new features of existing components, are used and object library usage is either impossible or undesirable, then the distribution problem can only be addressed by supplying the necessary shareable images to the end-user system.

See Section 1.6 for the requirements and setup needed for redistribution of the DEC C/C++ Run-Time Components kit.

1.1.3 Linking with Object Libraries

Linking with object libraries rather than shareable images eliminates the problem at end-user sites, but slows down the image activation process and consumes system resources.

When an OpenVMS VAX image is linked using object libraries instead of shareable images, all the code and local data for the application are linked directly into the resulting .EXE file (image). This makes the image independent of any shareable images on the linking system; therefore, you can ship the resulting image to any end-user site without the distribution problem occurring. Because the image includes all the necessary run-time code and data needed to execute the program, the image is larger than if shareable images were used.

You can link some (but not all) of the components on the DEC C/C++ Run-Time Components kit with object libraries. Specifically, the VAX C RTL is available in object library form (VAXCRTL.OLB, VAXCRTLG.OLB, VAXCCURSE.OLB), as is the DEC C RTL (DECCRTL.OLB, DECCRTLG.OLB, DECCCURSE.OLB). The other components on the DEC C/C++ Run-Time Components kit are available in shareable image form only. For information on linking with object libraries, see the *OpenVMS Linker Utility Manual* and the user's guides for VAX C and DEC C.

1.2 Saving the Old System Development Environment

As discussed in Section 1.1.1, one method of addressing the distribution problem is to save a copy of the existing link-time environment before installing the DEC C/C++ Run-Time Components kit. Software development and distribution sites often do this before upgrading OpenVMS VAX versions. This approach is simple, but does require sufficient additional disk space to hold the old files, and does require monitoring and deletion when the old environment is no longer required.

To save the old environment, find an alternative device and directory for the files. The alternative directory need not be on the system device. To determine the disk space required to store the old SYS$LIBRARY, enter the following command:

```
$ DIRECTORY/SIZE=ALL/TOTAL SYS$LIBRARY:*.*;0
```

Saving the existing environment to an alternative device is simple if sufficient disk space is available to hold all these files. For example, if the new directory is called DISK_D:[ALTSYSLIB], enter the following command to save the old environment:

```
$ COPY/LOG SYS$LIBRARY:*.*;0 DISK_D:[ALTSYSLIB]
```

To link against pre-DEC C/C++ Run-Time Components versions of the files in this saved environment, application build procedures only need to enter one of the following commands immediately before the appropriate LINK commands. If the LINK command uses SYS$LIBRARY, enter the following command:

```
$ DEFINE/USER SYS$LIBRARY DISK_D:[ALTSYSLIB]
```

If the LINK command uses SYS$SHARE, enter the following command:

```
$ DEFINE/USER SYS$SHARE DISK_D:[ALTSYSLIB]
```

If disk space is limited, you can store only those files replaced by the DEC C/C++ Run-Time Components kit. Doing this only requires about 1000 blocks but requires using a slightly more complicated logical name assignment at link time. To use the limited file storage method, enter the following COPY command before installing the DEC C/C++ Run-Time Components kit:

```
$ COPY/LOG SYS$LIBRARY:CMA$LIB_SHR.EXE,CMA$OPEN_LIB_SHR.EXE,-
CMA$OPEN_RTL.EXE,CMA$RTL.EXE,SMGSHR.EXE,VAXCRTL.EXE,VAXCRTLG.EXE -
DISK_D:[ALTSYSLIB]
```

This command does not place all the necessary files needed for linking applications in the alternative directory; therefore, you need a more complex logical name definition to link applications with these older files. If the LINK command uses SYS$LIBRARY, enter the following command:

```
$ DEFINE/USER SYS$LIBRARY DISK_D:[ALTSYSLIB],SYS$SYSROOT:[SYSLIB]
```

If the LINK command uses SYS$SHARE, enter the following command:

```
$ DEFINE/USER SYS$SHARE DISK_D:[ALTSYSLIB],SYS$SYSROOT:[SYSLIB]
```

This command ensures that the linker searches for all the components needed for the linking applications, first in the alternate directory and then in the standard directory. If your site has previously moved SYS$LIBRARY to an alternative disk, make sure you use that directory instead of SYS$SYSROOT:[SYSLIB].

1.3 Notifying Site Users

Before you can proceed to install the DEC C/C++ Run-Time Components kit, you need to:

- Perform the setup required to support the distribution of dependent products, including saving the DEC C/C++ Run-Time Components save set and documentation (see Section 1.6).
- Store the old SYS$LIBRARY in an alternative directory (see Section 1.2).

After the installation, notify all site users that the DEC C/C++ Run-Time Components kit has been installed. Warn users that if they are building applications for execution on other OpenVMS VAX Version 5.5, 5.5-1, or 5.5-2 systems they may encounter the distribution problem. Notify the users of the:

- Location of the saved DEC C/C++ Run-Time Components kit save set, this reference and installation manual, the read before installing cover letter, and the online release notes files provided by the installation procedure.
- Location of the alternate SYS$LIBRARY directory and the DEFINE command needed to use the components it contains.

In your application's release notes, refer to Chapter 1 of this guide so that software developers can take appropriate action to avoid causing problems for end users who execute on remote sites.

1.4 Diagnosing the Distribution Problem

If a software build procedure has been executed and the application image generated without any record (such as a .MAP file) of how it was linked, you can still determine if the image will encounter problems when the image executes at a remote site. Enter a single DCL command and consult Table 1–1 and Table 1–2. Using this method, software distributors can ensure that any applications they send to their user sites are free of DEC C/C++ Run-Time Components distribution problems.

To determine if an OpenVMS VAX executable or shareable image will have the distribution problem, enter the following command:

```
$ ANALYZE/IMAGE/OUTPUT=file-spec.ANL file-spec.EXE
```

file-spec is the file specification of the image to be examined. Examine the .ANL file with the TYPE command or an editor. At the end of the IMAGE HEADER section, you will see a series of image section descriptors (ISDs). Each ISD describes an image section. Image sections can come from object files, or shareable images that have been used during the LINK process.

Distinguishing such cases is easy: Image sections that come from shareable images appear at the end and contain additional information such as the global section name, match control, and the global section major and minor identifiers. By examining the list of global section names, and the identifiers, and comparing them with Table 1–1 and Table 1–2, you can determine which, if any, of the kit components have been referenced by the link operation.

Table 1–1 lists the new DEC C/C++ Run-Time Components that have not appeared on any OpenVMS VAX system before Version 6.0. If any of these global section names appear in the ISD list, then that image will encounter the distribution problem when executed at a remote site without the DEC C/C++ Run-Time Components kit installed.

Table 1–1 New Image Names Not Previously Available

Image Name	Major Identifier	Minor Identifier
CMA$TIS_SHR	1	1
CXXL$011_SHR_*nnn*[1]	1	1
DECC$SHR	2	0

[1] *nnn* is a version number, such as 001.

Table 1–2 lists the new DEC C/C++ Run-Time Components that replace existing OpenVMS Version 5.5, 5.5-1, or 5.5-2 shareable images and have their minor identifiers incremented. The major and minor identifiers are also listed. By comparing the names and identifiers in the .ANL file with those in Table 1–2, you can tell if any of the new versions were referenced. If there is no match in either table, then the image will not encounter startup problems due to the DEC C/C++ Run-Time Components kit distribution problem.

Table 1–2 New Image Names and Identifiers Replacing Existing OpenVMS Versions

Image Name	Major Identifier	Minor Identifier
VAXCRTL	4	5
VAXCRTLG	4	5

1.5 Symptoms and Appropriate Responses

If software developers on your system provide a target site with an image file (.EXE) linked against one of the kit components listed in Table 1–1 and Table 1–2, and that target site does not have the DEC C/C++ Run-Time Components kit installed, the image file will fail to activate. One of the following remedies must be applied:

1. Another version of the image file must be supplied that does not reference any of the kit component image files.
2. The DEC C/C++ Run-Time Components kit must be supplied, which makes the necessary components available.

The following messages indicate a distribution problem:

```
$ run xyz
%DCL-W-ACTIMAGE, error activating image VAXCRTL
-CLI-E-IMGNAME, image file SYS$SYSROOT:[SYSLIB]VAXCRTL.EXE;1
-SYSTEM-F-SHRIDMISMAT, ident mismatch with shareable image
```

VAXCRTL is one of the images listed in Table 1–2.

```
$ run xyz
%DCL-W-ACTIMAGE, error activating image CMA$TIS_SHR
-CLI-E-IMAGEFNF, image file not found SYS$SYSROOT:[SYSLIB]CMA$TIS_SHR.EXE;
```

CMA$TIS_SHR is one of the images listed in Table 1–1.

The site supplying the application VAX APP-XYZ is required to provide at least one of the two remedies. If the DEC C/C++ Run-Time Components kit has been installed without providing an alternative version of SYS$LIBRARY, then remedy 1 cannot be used. To use remedy 1, the older versions of the required files in SYS$LIBRARY must be recovered from a backup of your system device that was performed before installing the DEC C/C++ Run-Time Components kit.

If the DEC C/C++ Run-Time Components save set and documentation has not been saved on your system, then remedy 2 cannot be used. In this case, the original source for the DEC C/C++ Run-Time Components kit will need to be found. If you received the DEC C/C++ Run-Time Components kit with a Digital layered product, you will need to obtain the necessary save set and documentation from the media for that layered product. If you received the DEC C/C++ Run-Time Components kit from another supplier other than Digital, you will need to obtain it from them.

1.6 Requirements for Redistribution of the DEC C/C++ Run-Time Components Kit

The DEC C/C++ Run-Time Components kit may be redistributed along with any application that requires any of the components that are contained in it. For more information, see the DEC C/C++ Run-Time Components Software Product Description (SPD).

Note

The DEC C/C++ Run-Time Components kit and the components contained within it are supported only when the entire kit is redistributed to an end-user target site. Digital does not support redistribution of individual components contained in the kit.

To redistribute the DEC C/C++ Run-Time Components kit, you must supply the items listed in Section 1.7 to the end-user target site.

1.7 Saving the DEC C/C++ Run-Time Components Kit to Allow Redistribution

Digital has engineered the DEC C/C++ Run-Time Components kit specifically to allow redistribution with user-developed applications.

To facilitate redistribution, the DEC C/C++ Run-Time Components kit installer should ensure that copies of the following items are saved:

- A printed or binary copy of the read before installing cover letter for the DEC C/C++ Run-Time Components kit.

  The installation procedure puts the read before installing cover letter into the SYS$HELP directory. The name of the PostScript file is CRT060_RBI.PS; the name of the text file is CRT060_RBI.TXT.

- A printed or binary copy of the reference and installation guide for the DEC C/C++ Run-Time Components kit.

  The installation procedure puts the reference and installation guide into the SYS$HELP directory. The name of the PostScript file is CRT060_REF_INST.PS; the name of the text file is CRT060_REF_INST.TXT.

- The complete DEC C/C++ Run-Time Components kit save set, AACRT060.A.

Save these items immediately after installing the DEC C/C++ Run-Time Components kit. These items can either be saved online for easy access or on a separate medium that is available and appropriately labeled. You can copy the binary items from the media they are provided on with the DCL command COPY or BACKUP.

If you do not save the DEC C/C++ Run-Time Components kit save set, it cannot be reconstituted later from the installed components.

2
Preparing for Installation

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.

DEC C/C++ Run-Time Components Version 6.0 for OpenVMS VAX provides online release notes, which are copied to the following files during installation (*n* is the minor release number):

```
SYS$HELP:CRT06n.RELEASE_NOTES
SYS$HELP:CRT06n_RELEASE_NOTES.PS
SYS$HELP:CRT06n_RELEASE_NOTES.DECW$BOOK
```

Digital strongly recommends that you read the release notes before proceeding with the installation. For information on accessing the online release notes, see step 5 in Section 3.1.

2.1 Required Operating System Components and Software

DEC C/C++ Run-Time Components Version 6.0 for OpenVMS VAX requires the OpenVMS VAX operating system Version 5.5 or higher. Future DEC C/C++ Run-Time Components Version 6.0 for OpenVMS VAX releases may require higher versions of the operating system, as described in the online release notes.

The OpenVMS VAX operating system comes with a variety of support options or components. Components include such features as networking and RMS journaling. To use DEC C/C++ Run-Time Components Version 6.0 for OpenVMS VAX, your system should be running a version of OpenVMS VAX that includes the base component and the following:

- Programming support
- Utilities

The run-time component kit supplies versions of several OpenVMS VAX component files that are more recent than versions supplied as part of the OpenVMS VAX upgrades for Version 5.5 through Version 5.5-2. After installing this kit, if you then upgrade your system to one of these intermediate versions, you will have to reinstall the component kit. Otherwise, applications that use new features of these components may fail to work properly.

2.2 Installation Procedure Requirements

This section discusses the settings to check to ensure that your system is ready for the installation of the DEC C/C++ Run-Time Components to proceed.

2.2.1 Process Privileges

To install the DEC C/C++ Run-Time Components, you must be logged in to an account that has SETPRV or at least the following privileges:

- CMKRNL
- WORLD
- SYSPRV

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

2.2.2 Process Account Quotas

VMSINSTAL requires that the installation account have minimum quotas, as shown in Table 2–1.

Table 2–1 Process Quotas for the Installing Account

ASTLM	1000
BIOLM	1000
BYTLM	99,000
DIOLM	1000
ENQLM	4000
FILLM	300

Use the OpenVMS Authorize Utility to verify and change process quotas for the installation account in the user authorization file (UAF). (Some sites may restrict the use of the OpenVMS Authorize Utility to certain accounts or people.) For instance, to change the BYTLM quota for the *account-name* installation account, you might enter the following command sequence:

```
$ RUN SYS$SYSTEM:AUTHORIZE
UAF> MODIFY account-name /BYTLM = 99000
UAF> SHOW account-name
UAF> EXIT
$ LOGOUT
```

After the quotas for the installation account have been changed, log out of the installation account and log in again for the new quotas to take effect. You can then proceed with the installation.

User account quotas are stored in the file SYSUAF.DAT. For more information on modifying account quotas, see the description of the Authorize Utility in the OpenVMS System Management Subkit.

2.2.3 System Parameters

Table 2–2 lists the minimum required system parameter values for the installation. Depending on the kinds of programs and applications running at your site, you might need higher values for some settings.

Table 2–2 Minimum Required System Parameter Values

System Parameter	Minimum Value	Dynamic Value
OpenVMS VAX Version 6.0		
CLISYMTBL[1]	500	500
GBLPAGES[2]	200	200
GBLPAGFIL	200	200
Global Sections[2]	10	10
OpenVMS VAX Versions 5.5, 5.5-1, and 5.5-2		
CLISYMTBL[1]	500	500
GBLPAGES[2]	1500	1500

[1]The CLISYMTBL dynamic system parameter must be set to a minimum value of 500 during the installation procedure. You can change the setting to its original value after the installation is finished.

[2]These values 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.

(continued on next page)

Table 2–2 (Cont.) Minimum Required System Parameter Values

System Parameter	Minimum Value	Dynamic Value
OpenVMS VAX Versions 5.5, 5.5-1, and 5.5-2		
GBLPAGFIL	1500	1500
Global Sections[2]	33	33

[2]These values 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.

2.2.4 Disk Space

The DEC C/C++ Run-Time Components Version 6.0 for OpenVMS VAX requirements for free disk storage space are different during installation and after installation. Table 2–3 summarizes the storage requirements.

Table 2–3 Disk Space Requirements

Kit	Blocks During Installation	Blocks After Installation
OpenVMS VAX Version 6.0		
DEC C/C++ Run-Time Components Version 6.0 for OpenVMS VAX	13,000	2000
Installation Verification Procedures	150	50
OpenVMS VAX Versions 5.5, 5.5-1, and 5.5-2		
DEC C/C++ Run-Time Components Version 6.0 for OpenVMS VAX	17,000	6000
Installation Verification Procedures	150	50

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
```

2.2.5 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 back up your system disk before installing any software.

Use the backup procedures that are established at your site. For details on performing a system disk backup, see the section on the Backup Utility in the OpenVMS System Management Subkit.

2.2.6 Additional System Maintenance Procedures

This section explains how to use the System Generation Utility (SYSGEN) and the AUTOGEN Utility to prepare your system for an installation of the DEC C/C++ Run-Time Components. AUTOGEN uses system configuration data to automatically set system parameters.

2.2.6.1 Checking System Parameter Values

To check the values of your system parameters, enter the following command:

```
$ RUN SYS$SYSTEM:SYSGEN
SYSGEN>
```

At the SYSGEN> prompt, enter the SHOW command to display the value of a system parameter. The following example displays the value for the CLISYMTBL system parameter:

```
SYSGEN> SHOW CLISYMTBL
```

After checking the parameters with the SHOW command, enter the EXIT command at the SYSGEN> prompt to return to DCL level. For information on setting dynamic system parameter values, see Section 2.2.6.4.

2.2.6.2 Calculating the Values for Global Pages and Global Sections

To install and run DEC C/C++ Run-Time Components Version 6.0 for OpenVMS VAX, 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 global pages and global sections system parameters.

Enter the following DCL command to determine the number of global pages required by SYS$LIBRARY:DCLTABLES.EXE:

```
$ DIR/SIZE SYS$LIBRARY:DCLTABLES.EXE
```

This command returns the size (in blocks) of SYS$LIBRARY:DCLTABLES.EXE. As an approximation, use this figure as the number of global pages for the file.

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

If the values displayed by the system are greater than the values in Table 2–2, you do not need to increase the values for these parameters. If the value of free global pages or global sections is less than the value in Table 2–2, you must increase the system parameter setting.

Section 2.2.6.3 describes the procedures for increasing these values using AUTOGEN.

2.2.6.3 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
```

Use an editor to access the file. 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 = 8096
```

To modify incremental parameters such as free global pages and global sections, prefix GBLPAGES and GBLSECTS with ADD_. The following example increases the global page setting by 2000:

```
ADD_GBLPAGES = 2000
```

When you set the page file quota, do not 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. The following command recalculates your system parameters and reboots the system:

```
$ @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.

AUTOGEN 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 your operating system.

2.2.6.4 Setting Dynamic System Parameter Values

Use SYSGEN to set dynamic parameters. Dynamic parameters changed with the SYSGEN command WRITE ACTIVE become active immediately without any need to reboot your system. Rebooting returns dynamic system parameter values to their previous settings.

After you change dynamic parameter values, complete the installation before rebooting the system. After you finish the installation, you can reset the dynamic parameters to their previous values or let them be reset automatically when you reboot your system the next time.

If the dynamic parameter values on your system are less than the values previously listed, use the following series of commands to change the values. This example changes the CLISYMTBL value to 500.

```
$ RUN SYS$SYSTEM:SYSGEN
SYSGEN> USE ACTIVE
SYSGEN> SET CLISYMTBL 500
SYSGEN> WRITE ACTIVE
SYSGEN> EXIT
```

3

Installing the DEC C/C++ Run-Time Components

The installation takes approximately 3 to 10 minutes, depending on your system configuration and the kit components chosen.

3.1 Installation Procedure

The DEC C/C++ Run-Time Components Version 6.0 for OpenVMS VAX installation procedure consists of a series of questions and informational messages. See Appendix A for a sample installation.

To abort the installation procedure at any time, press Ctrl/Y. When you press Ctrl/Y, the installation procedure deletes all the files 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 ([ ]).

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

   ```
   Username: SYSTEM
   Password:
   ```

2. **Invoke VMSINSTAL.**

   ```
   @SYS$UPDATE:VMSINSTAL saveset-name device-name option-list
   ```

   To start the installation, invoke the VMSINSTAL command procedure from a privileged account, such as the SYSTEM account. VMSINSTAL is in the SYS$UPDATE directory. The elements that you specify when you invoke the VMSINSTAL command procedure are as follows:

saveset-name

The installation name for the component. For DEC C/C++ Run-Time Components Version 6.0 for OpenVMS VAX, use the following installation name:

AACRT06*n*

n represents the minor (point) release number for DEC C/C++ Run-Time Components Version 6.0 for OpenVMS VAX.

device-name

The name of the device on which you plan to mount the media and the directory containing the DEC C/C++ Run-Time Components Version 6.0 for OpenVMS VAX save set. For example, DSK$: is the device name used in examples in this guide. The directory used in examples in this guide is [CRT06*n*], where *n* is the minor release number. The actual directory name may differ because the DEC C/C++ Run-Time Components may be included with another product's save set. If this is the case, for the correct directory specification, see the installation guide for the product that includes the DEC C/C++ Run-Time Components save set.

To specify the device name on the @VMSINSTAL command line, enter:

DSK$:[CRT06*n*]

option-list

The word OPTIONS followed by one or more of the following option designators—A, G, L, N, R:

- Auto_answer option (A)

  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).

- Get save-set option (G)

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

- File log option (L)

  Logs all activity to the terminal during installation.

- Display or print the release notes (N)

  Indicates that you want to see the installation question on release notes. If you do not include the N option, 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 the N option. (Even if you omit the N option when you invoke VMSINSTAL, the DEC C/C++ Run-Time Components Version 6.0 for OpenVMS VAX release notes are copied automatically to the SYS$HELP directory.)

- Alternate root option (R)

  Lets you install the product to a system root other than that of the running system.

For detailed information on these software installation options, see the OpenVMS VAX operating system documentation. If you specify more than one option, separate the options with commas, such as OPTIONS A,N.

The following example invokes VMSINSTAL to install DEC C/C++ Run-Time Components Version 6.0 for OpenVMS VAX and shows the system response. This example uses the OPTIONS N release note option.

```
$ @SYS$UPDATE:VMSINSTAL AACRT06n DSK$:[CRT06n] OPTIONS N

      OpenVMS VAX Software Product Installation Procedure V5.5

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

The following example invokes VMSINSTAL to install DEC C/C++ Run-Time Components Version 6.0 for OpenVMS VAX from tape drive MTA0: and shows the system response. This example uses the OPTIONS N release note parameter.

```
$ @SYS$UPDATE:VMSINSTAL AACRT06n MTA0: OPTIONS N

      OpenVMS VAX Software Product Installation Procedure V5.5

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 this information later in the installation procedure. VMSINSTAL does not prompt you for any options, so be sure to include OPTIONS N on the VMSINSTAL command line to access the release notes during the installation.

When you invoke VMSINSTAL, it checks the following:

- Whether you are 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 quota values shown in Table 2–1.

VMSINSTAL then checks to determine whether any users are logged in to the system. If users are logged in, VMSINSTAL asks whether you want to continue the installation:

```
* Do you want to continue anyway [NO]?  YES
```

If you want to continue, enter YES. If you want to stop the installation, press Return.

3. **Confirm system backup.**

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

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.

If you omitted the device name on the VMSINSTAL command line, the following prompt appears:

```
* Where will the distribution volumes be mounted:
DSK$:[CRT06n]
```

If this prompt appears, enter the name of the distribution (media) device and directory that contains the DEC C/C++ Run-Time Components Version 6.0 for OpenVMS VAX kit (save set). For example, if the media is contained on a CD-ROM located on unit DSK$:, enter DSK$:[CRT06*n*] in reply to this question (where *n* is the minor version number).

You should now mount the first distribution volume on the device you specified when you invoked VMSINSTAL.

Depending on the type of device and whether the device is already mounted, the following prompt may or may not appear:

```
Please mount the first volume of the set on DSK$:.
* Are you ready? YES
%MOUNT-I-MOUNTED, DSK$:  mounted
```

If you have not already done so, insert the distribution volume on the device you specified when you invoked VMSINSTAL. The device name appears in the line preceding the question. VMSINSTAL then asks 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 C/C++ Run-Time Components Version 6.0 for OpenVMS VAX 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.

4. **Verify the product being installed**.

```
 The following products will be processed:

   AACRT V6.n

     Beginning installation of AACRT V6.n  at hh:mm

   %VMSINSTAL-I-RESTORE, Restoring product save set A...
```

The product name and version are displayed. This save set (A) is restored to a subdirectory of SYS$UPDATE:.

5. **Select a release notes option.**

```
Release Notes Options:

 1.  Display release notes
 2.  Print release notes
 3.  Both 1 and 2
 4.  Copy release notes to SYS$HELP

* Select option [2]:
```

If you specified OPTIONS N when you invoked VMSINSTAL, you are now asked to choose one of the options for reviewing the release notes. All options copy the release notes to SYS$HELP.

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. Either supply the name of the print queue that you want to use or press Return to send the file to the default output print device. For example:

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

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.

If you select option 4, the installation procedure will only copy the release notes to the SYS$HELP directory.

Next, VMSINSTAL displays the following message:

```
%VMSINSTAL-I-RELMOVED, Product's release notes have been moved to SYS$HELP.

         DEC C/C++ Run-Time Components V6.n for OpenVMS VAX Systems

  Copyright (c) Digital Equipment Corporation, 1993.  All rights reserved.
```

The release notes are copied to the following files in the SYS$HELP directory (where *n* is the minor version number):

```
SYS$HELP:CRT06n.RELEASE_NOTES
SYS$HELP:CRT06n_RELEASE_NOTES.PS
SYS$HELP:CRT06n_RELEASE_NOTES.DECW$BOOK
```

Note

The name of the release notes file (installed by VMSINSTAL) consists of the current product name and version number.

6. **Heed warning message about image dependency.**

   The installation procedure now displays the following message:

```
This kit provides new versions of several OpenVMS VAX standard
shareable images and object libraries in SYS$LIBRARY, such as
VAXCRTL.EXE, CMA$RTL.EXE, and SMGSHR.EXE. These provide support for
several programming capabilities provided by OpenVMS VAX Version 6.0
for C, C++, and other languages.

******************************************************************
*                         W A R N I N G                          *
******************************************************************
```

```
Programs  that link  against the new  versions of any of  these files,
EVEN THOSE NOT WRITTEN IN DEC C OR C++, will NOT run on older versions
of OpenVMS (V5.5-2 or earlier)  unless steps are taken to save the old
SYS$LIBRARY configuration or redistribute this kit to additional sites.
See Chapter 1 of the DEC C/C++ Run-Time Components for OpenVMS VAX
Reference and Installation Guide for more details.

In  addition,  if you  update the system  version of  OpenVMS VAX, the
enhanced  versions of the new images and libraries may be overwritten
with  versions that lack the  enhanced support.  Be sure to reinstall
the DEC C/C++ Run-Time Components kit after each OpenVMS VAX system
update.
```

VMSINSTAL now asks if you wish to continue with the installation.

```
* Continue with the installation? [N]? YES
```

If you have adequately saved your old system development environment as described in Section 1.2, enter YES and press Return. If you have not saved the old environment, you can abort the installation by pressing Return, save the environment, and then restart the installation.

7. **Purge files.**

   VMSINSTAL now asks you if you want to purge files replaced by the installation procedure.

   ```
   * Do you want to purge files replaced by this installation [YES]? YES
   ```

   The directories SYS$HELP, SYS$LIBRARY, SYS$MESSAGE, and SYS$TEST may contain previous versions of DEC C/C++ Run-Time Components files. These files are replaced during the installation with new files, but the old files are not automatically purged. In response to the purging prompt, press Return or enter YES and press Return to purge the files. To keep the files, enter NO and press Return.

8. **Choose the Installation Verification Procedure option.**

   VMSINSTAL now asks if you want to run the Installation Verification Procedure (IVP). The IVP checks to be sure that the installation is successful. Digital recommends that you run the IVP.

   ```
   This kit contains an Installation Verification Procedure (IVP) to verify
   the correct installation of the DEC C/C++ Run-Time Components.  The IVP
   will be placed in:

      SYS$TEST:CRT$IVP.COM.
   ```

```
After the installation is complete, you can invoke the command file
at any time to reverify that the DEC C/C++ Run-Time Components are
installed and working correctly. Answer YES to the next question
if you wish to run the IVP at the completion of this installation
procedure.

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

After DEC C/C++ Run-Time Components Version 6.0 for OpenVMS VAX is installed, 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 C/C++ Run-Time Components Version 6.0 for OpenVMS VAX.

9. **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 files to their target directories. The new files include the following:

```
SYS$HELP:CRT06n.RELEASE_NOTES                       [new]
SYS$HELP:CRT06n.RELEASE_NOTES.PS                    [new]
SYS$HELP:CRT06n.RELEASE_NOTES.DECW$BOOK             [new]
SYS$HELP:CRT060_REF_INST.PS                         [new]
SYS$HELP:CRT060_REF_INST.TXT                        [new]
SYS$HELP:CRT060_RBI.PS                              [new]
SYS$HELP:CRT060_RBI.TXT                             [new]
SYS$TEST:CRT$IVP.COM                                [new]
SYS$TEST:[CRT]CRT$ERR1.OBJ                          [new]
SYS$TEST:[CRT]CRT$ERR2.OBJ                          [new]
SYS$TEST:[CRT]CRT$IVP1.OBJ                          [new]
SYS$TEST:[CRT]CRT$IVP2.OBJ                          [new]
SYS$TEST:[CRT]CRT$IVP3.OBJ                          [new]
SYS$TEST:[CRT]CRT$IVP4.OBJ                          [new]
```

   Also, if you answered YES in response to the prompt that requests a file purge at step 7, previous versions of the DEC C/C++ Run-Time Components files are now purged.

   Additional components follow that are the primary functional components supplied by the kit. The actual functional components supplied will differ depending on: the OpenVMS VAX version your system is running, the layered products already installed on your system, and whether or not this is a reinstallation.

VMSINSTAL displays the following reminder to invoke the CRT$STARTUP.COM command procedure:

```
******************************************************************
*                        R E M I N D E R                         *
******************************************************************

If you are  installing on a  VMScluster, you  MUST execute the command
file  SYS$STARTUP:CRT$STARTUP.COM on all members of the  cluster after
installing the DEC C/C++ Run-Time Components kit.

If you are  installing on an OpenVMS VAX system  prior to Version 6.0,
ensure that you invoke the  SYS$STARTUP:CRT$STARTUP.COM  startup  file
prior to  starting up any other  products in the system  startup file,
SYS$MANAGER:SYSTARTUP_V5.COM.  See Chapter 4 of the DEC C/C++ Run-Time
Components for OpenVMS VAX  Reference and Installation Guide  for more
details on postinstallation operations.
```

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

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

10. **Observe the IVP.**

    If you chose to run the IVP (step 8), VMSINSTAL now invokes the IVP to verify that DEC C/C++ Run-Time Components Version 6.0 for OpenVMS VAX was installed successfully.

    If DEC C/C++ Run-Time Components Version 6.0 for OpenVMS VAX was installed successfully, the IVP displays the following:

```
** DEC C/C++ Run-Time Components Installation Verification Procedure **
** Ending at hh:mm **
```

11. **End the installation procedure.**

```
Installation of AACRT V6.n completed at hh:mm

VMSINSTAL procedure done at hh:mm
```

    Depending on whether you specified parameters on the VMSINSTAL command line, the * Products: prompt or the DCL prompt ($) may appear. If the * Products: prompt appears, you can return to the DCL prompt by pressing Ctrl/Z.

    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 installation account (such as SYSTEM) and you want to restore these symbols, you should log out and log in again.

3.2 Error Recovery

If VMSINSTAL detects any problems during the installation, it notifies you and asks if you want to continue the installation. In some instances, you can enter YES to continue. To stop the installation process and correct the situation, enter NO or press Return. Correct the problem and restart the installation.

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 will see the following message:

```
%VMSINSTAL-E-INSFAIL, DEC C/C++ Run-Time Components Installation Aborted.
```

If the IVP fails, you will see this message:

```
** DEC C/C++ Run-Time Components Installation Verification 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 VAX documentation on system messages, recovery procedures, and OpenVMS VAX 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 2.

If the installation fails, you must restart the installation procedure from step 2. For an explanation of any system messages, see the *OpenVMS System Messages and Recovery Procedures Reference Manual* or the online HELP/MESSAGE facility. If the installation fails due to an IVP failure, contact a Digital field service representative.

4
After Installation

DEC C/C++ Run-Time Components Version 6.0 for OpenVMS VAX is now installed and can be used by all applications written in DEC C or DEC C++, or that use the DECthreads facility.

The following tasks can be performed after DEC C/C++ Run-Time Components Version 6.0 for OpenVMS VAX is installed:

- Running the Installation Verification Procedure (IVP) separately
- Editing the system startup file
- Making the product usable on a VMScluster
- Reinstalling after an OpenVMS VAX intermediate upgrade
- Reporting problems

The following sections describe these tasks in detail.

4.1 Running the Installation Verification Procedure Separately

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

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

If an IVP fails for any reason, the following message is displayed:

```
** DEC C/C++ Run-Time Components Installation Verification FAILED **
```

4.2 Editing the System Files

To use DEC C/C++ Run-Time Components Version 6.0 for OpenVMS VAX with DECwindows or other applications written in the C language, you need to edit the system startup file to provide for automatic inclusion of the DEC C/C++ Run-Time Components as known images when your system is rebooted. This step is required only on OpenVMS VAX systems prior to Version 6.0.

Add the command line that starts the DEC C/C++ Run-Time Components kit to the system startup file, SYS$MANAGER:SYSTARTUP_V5.COM. You should position this new command line after the line that invokes the network startup command procedure. The following example shows the network startup command line followed by the DEC C/C++ Run-Time Components Version 6.0 for OpenVMS VAX startup command line:

```
$ @SYS$MANAGER:STARTNET.COM
      .
      .
      .
$ @SYS$STARTUP:CRT$STARTUP.COM
```

Note

This line should also precede the startup invocation of any applications written in the C or C++ languages, or that use the DECthreads facility. Examples of such applications are the DEC TCP/IP Services for OpenVMS (UCX$STARTUP.COM), DEC DCE, and DECmcc. This startup file must be invoked on all OpenVMS VAX nodes in a VMScluster.

4.3 VMScluster Considerations

After completing the initial installation on OpenVMS VAX systems prior to Version 6.0, you must install the DEC C/C++ Run-Time Components kit on all nodes in a cluster. You must run the startup on each of the other nodes and enter the following command:

```
$ @SYS$STARTUP:CRT$STARTUP.COM
```

If you are using a dual-architecture (VAX and AXP systems) heterogeneous cluster, make sure the appropriate logical names have been set to define the scope of the SYSMAN DO commands before executing these commands. For more information, see the *OpenVMS System Manager's Manual*.

4.4 Reinstalling After an OpenVMS VAX Intermediate Upgrade

If you upgrade your operating system to OpenVMS VAX Version 5.5-1 or 5.5-2 after installing this kit, you must reinstall the DEC C/C++ Run-Time Components kit. Unless you do reinstall this kit, applications that use new features of the DEC C/C++ Run-Time Components kit may fail to work properly, because the kit supplies versions of several OpenVMS VAX component files that are more recent than versions supplied as part of the previous intermediate upgrades.

4.5 Reporting Problems

If you encounter a problem while using the DEC C/C++ Run-Time Components Version 6.0 for OpenVMS VAX kit, report it to Digital. Depending on the type of support you have and the nature of the problem, take one of the following actions:

- If you have a Software Product Services Support Agreement, contact your Digital Customer Support Center (CSC), either by telephone or using the electronic means provided with your support agreement (such as DSNlink). The CSC provides telephone support for high-level advisory and remedial assistance. When you initially contact the CSC, please indicate the following:
  - The name (OpenVMS VAX) and the version number of the operating system you are using.
  - The product name (DEC C/C++ Run-Time Components Version 6.0 for OpenVMS VAX) and the version number of DEC C/C++ Run-Time Components Version 6.0 for OpenVMS VAX you are using.
  - The hardware system you are using, such as a model number.
  - A very brief description of the problem (one sentence if possible).
  - How critical the problem is.

  When you submit information electronically or are speaking on the phone to the appropriate DEC C/C++ Run-Time Components Version 6.0 for OpenVMS VAX support specialist, you can provide more detailed information. This includes the specific commands used to compile and link the program, the error messages displayed, and relevant detailed information (possibly including source program listings). Please attempt to narrow the cause of the problem to a specific module or lines of code.

CSC personnel may ask for additional information, such as listings of any command files, INCLUDE files, relevant data files, and so forth. If the program is longer than 50 lines, submit a copy of it electronically or provide machine-readable media (floppy diskette or magnetic tape).

- If you have a Self-Maintenance Software Agreement, you can submit a Software Performance Report (SPR).

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

1. Describe as accurately as possible the circumstances and state of the system when the problem occurred. In the description, please provide:
   - The version number of DEC C/C++ Run-Time Components Version 6.0 for OpenVMS VAX being used.
   - The name and version number of the operating system being used.
   - A concise description of the problem with specific examples. Include any error messages displayed, the commands used to compile and link the program, and so on.
2. Reduce the problem to as small a size as possible.
3. Provide a listing of the program.
4. When you submit an SPR, remember to include listings of any command files, INCLUDE files, relevant data files, and so forth.
5. When you submit an SPR, if the program is longer than 50 lines, submit a copy of it on machine-readable media (floppy diskette or magnetic tape). If necessary, also submit a copy of the program library used to build the application.
6. Report only one problem per SPR. This will facilitate a faster response.
7. Mail the SPR package to Digital.

Many problem reports do not contain enough information to duplicate or identify the problem. Concise, complete information helps Digital give you accurate and timely service to software problems.

To obtain information about purchasing a Digital Software Product Services Support Agreement or other support services, please contact your local Digital sales representative.

A
Sample Installation

This appendix contains a sample installation for the DEC C/C++ Run-Time Components Version 6.0 for OpenVMS VAX kit. This is only a sample. Sometimes the installation procedure is modified in maintenance updates of this product.

The following sample was installed on a system where DEC C/C++ Run-Time Components Version 6.0 for OpenVMS VAX was not previously installed. Depending on which layered products you have on your system, you might see additional messages and questions when you perform your installation.

This 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.

Example A–1 shows a sample installation of the DEC C/C++ Run-Time Components on an OpenVMS VAX system.

Example A–1 Sample DEC C/C++ Run-Time Components Installation on an OpenVMS VAX System

```
        Welcome to VAX/VMS V5.5-2

Username: SYSTEM
Password:
        Welcome to VAX/VMS version V5.5-2 on node SUBTLE
    Last interactive login on Thursday, 19-AUG-1993 14:39
    Last non-interactive login on Monday,  9-AUG-1993 15:48

$ @sys$update:vmsinstal aacrt060 sys$update

VAX/VMS Software Product Installation Procedure V5.5-2

It is 19-AUG-1993 at 14:43.

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

* Are you satisfied with the backup of your system disk [YES]?
* Enter installation options you wish to use (none):

The following products will be processed:
  AACRT V6.0

Beginning installation of AACRT V6.0 at 14:43

%VMSINSTAL-I-RESTORE, Restoring product save set A ...
%VMSINSTAL-I-RELMOVED, Product's release notes have been moved to SYS$HELP.
  DEC C/C++ Version V06.0-60 Run-Time components for OpenVMS/VAX systems

    Copyright (c) Digital Equipment Corporation, 1993.  All rights reserved.

    This kit  provides  new  versions  of  several  OpenVMS  VAX  standard
    shareable  images  and  object  libraries  in  SYS$LIBRARY,  such  as
    VAXCRTL.EXE,  CMA$RTL.EXE, and  SMGSHR.EXE.  These provide support for
    several programming  capabilities provided by  OpenVMS VAX Version 6.0
    for C, C++, and other languages.

    ******************************************************************
    *                          W A R N I N G                         *
    ******************************************************************

    Programs  that link  against the new  versions of any of  these files,
    EVEN THOSE NOT WRITTEN IN DEC C OR C++, will NOT run on older versions
    of OpenVMS (V5.5-2 or earlier)  unless steps are taken to save the old
    SYS$LIBRARY configuration or redistribute this kit to additional sites.
    See Chapter 1 of the  DEC C/C++  Run-Time Components  for OpenVMS  VAX
    Reference and Installation Guide for more details.
```

(continued on next page)

Example A–1 (Cont.) Sample DEC C/C++ Run-Time Components Installation on an OpenVMS VAX System

```
    In  addition,  if you  update the system  version of  OpenVMS VAX, the
    enhanced  versions of the new images and libraries may be  overwritten
    with  versions that lack the  enhanced support.  Be sure to  reinstall
    the DEC C/C++ Run-Time Components kit after each OpenVMS VAX  system
    update.

* Continue with the installation? [N]? y

* Do you want to purge files replaced by this installation [YES]? n

    This kit  contains an  Installation  Verification  Procedure  (IVP) to
    verify the correct installation of the DEC C/C++ Run-Time Components.
    The IVP will be placed in:

        SYS$TEST:CRT$IVP.COM

    After the installation is complete, you can invoke the command file at
    any time to reverify that DEC C/C++ Run-Time Components are installed
    and working correctly.  Answer yes to the next question if you wish to
    run the IVP at the completion of this installation procedure.

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

        All questions and verifications regarding the installation of the
        DEC C/C++ Run-Time Components have been completed.  Depending on
        your configuration, time estimates for the remainder of the
        installation are 1 to 3 minutes.

%VMSINSTAL-I-SYSDIR, This product creates system directory [SYSTEST.CRT].

  Your OpenVMS system will now be updated to include the following new file(s):

    SYS$HELP:CRT060.RELEASE_NOTES                       [new]
    SYS$HELP:CRT060_RELEASE_NOTES.PS                    [new]
    SYS$HELP:CRT060_RELEASE_NOTES.DECW$BOOK             [new]
    SYS$HELP:CRT060_REF_INST.PS                         [new]
    SYS$HELP:CRT060_REF_INST.TXT                        [new]
    SYS$HELP:CRT060_RBI.PS                              [new]
    SYS$HELP:CRT060_RBI.TXT                             [new]
    SYS$TEST:CRT$IVP.COM                                [new]
    SYS$TEST:[CRT]CRT$ERR1.OBJ                          [new]
    SYS$TEST:[CRT]CRT$ERR2.OBJ                          [new]
    SYS$TEST:[CRT]CRT$IVP1.OBJ                          [new]
    SYS$TEST:[CRT]CRT$IVP2.OBJ                          [new]
    SYS$TEST:[CRT]CRT$IVP3.OBJ                          [new]
    SYS$TEST:[CRT]CRT$IVP4.OBJ                          [new]
```

(continued on next page)

Example A–1 (Cont.) Sample DEC C/C++ Run-Time Components Installation on an OpenVMS VAX System

```
    SYS$STARTUP:CRT$STARTUP.COM                          [new]
    SYS$LIBRARY:CMA$RTL.EXE                              [new]
    SYS$LIBRARY:CMA$OPEN_RTL.EXE                         [new]
    SYS$LIBRARY:CMA$LIB_SHR.EXE                          [new]
    SYS$LIBRARY:CMA$OPEN_LIB_SHR.EXE                     [new]
    SYS$LIBRARY:CMA$TIS_SHR.EXE                          [new]
    SYS$LIBRARY:CXXL$011_SHR.EXE                         [new]
    SYS$MESSAGE:CXXL$MSG_SHR.EXE                         [new]
    SYS$LIBRARY:DECC$SHR.EXE                             [new]
    SYS$LIBRARY:DECCCURSE.OLB                            [new]
    SYS$LIBRARY:DECCRTL.OLB                              [new]
    SYS$LIBRARY:DECCRTLG.OLB                             [new]
    SYS$LIBRARY:UCX$CRTLIBXFR.OBJ                        [new]
    SYS$LIBRARY:VAXC$EMPTY.EXE                           [new]
    SYS$LIBRARY:VAXC$LCL.OPT                             [new]
    SYS$LIBRARY:VAXC2DECC.EXE                            [new]
    SYS$LIBRARY:VAXCRTL.EXE                              [new]
    SYS$LIBRARY:VAXCG2DECC.EXE                           [new]
    SYS$MESSAGE:VAXCMSG.EXE                              [new]
    SYS$LIBRARY:VAXCCURSE.OLB                            [new]
    SYS$LIBRARY:VAXCRTL.OLB                              [new]
    SYS$LIBRARY:VAXCRTLG.EXE                             [new]
    SYS$LIBRARY:VAXCRTLG.OLB                             [new]
    SYS$LIBRARY:DECC$EMPTY.EXE                           [new]

    ******************************************************************
    *                        R E M I N D E R                         *
    ******************************************************************

    If you are  installing on a  VMScluster, you  MUST execute the command
    file  SYS$STARTUP:CRT$STARTUP.COM on all members of the  cluster after
    installing the DEC C/C++ Run-Time Components kit.

    If you are  installing on an OpenVMS VAX system  prior to Version 6.0,
    ensure that you invoke the  SYS$STARTUP:CRT$STARTUP.COM  startup  file
    prior to  starting up any other  products in the system  startup file,
    SYS$MANAGER:SYSTARTUP_V5.COM.  See Chapter 4 of the DEC C/C++ Run-Time
    Components  for OpenVMS VAX  Reference and Installation Guide for more
    details on postinstallation operations.

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

  Copyright (c) Digital Equipment Corporation, 1993.  All rights reserved.

    *-----------------------------------------*
    *    Installation Verification Procedure    *
    *     for DEC C/C++ Run-Time Components *
    *-----------------------------------------*
```

(continued on next page)

Example A–1 (Cont.) Sample DEC C/C++ Run-Time Components Installation on an OpenVMS VAX System

```
   *-----------------------------------------*
   *    Test that components LINK properly     *
   *-----------------------------------------*

   *-----------------------------------------*
   *    Test that components RUN properly      *
   *-----------------------------------------*

   Testing DEC C RTL . . .
   Testing DEC C++ Class Library . . .
   Testing curses . . .
   Testing DEC C RTL . . .
   Testing DEC C++ Class Library . . .
   Testing curses . . .
       Testing VAX C RTL . . .
       Testing curses . . .
       Testing DECthreads . . .
       Testing VAX C RTL . . .
       Testing curses . . .
       Testing DECthreads . . .
       Testing VAX C RTL . . .
       Testing curses . . .
       Testing DECthreads . . .
       Testing VAX C RTL . . .
       Testing curses . . .
       Testing DECthreads . . .
       Testing VAX C RTL . . .
       Testing curses . . .
       Testing DECthreads . . .
       Testing VAX C shareable usage . . .
       Testing shareable portion . . .
       Testing main portion . . .
       Testing VAX C shareable usage . . .
       Testing shareable portion . . .
       Testing main portion . . .

** DEC DEC C/C++ Run-Time Components Installation Verification Procedure **
** Ending at 19-AUG-1993 14:47:05.63 **

      Installation of AACRT V6.0 completed at 14:47

      VMSINSTAL procedure done at 14:50
```