DIGITAL PROPRIETARY INFORMATION

RELEASE NOTES

Version 1.3B of Digital DCE for OpenVMS VAX and OpenVMS Alpha

Version 1.3B Mandatory Update

Version 1.3B of Digital DCE for OpenVMS VAX and OpenVMS Alpha is a mandatory update that replaces Digital DCE for OpenVMS VAX and OpenVMS Alpha Version 1.3A. *Version 1.3B is a complete kit that does not require a previous version of Digital DCE for OpenVMS for installation.* Version 1.3B can be installed in a new system or can be installed as an update to a previous version of DCE for OpenVMS.

All existing Version 1.3 documentation can be used with Version 1.3B with the following exceptions:

- References to the installation directories [DCEVAX013] and [DCEAXP013] should be changed to [DCEVAXMUPB013] and [DCEAXPMUPB013].
- References to the installation saveset names DCEVAX013 and DCEAXP013 should be changed to DCEVAXMUPB013 and DCEAXPMUPB013.

1 Services Digital DCE Offers

Version 1.3B of Digital DCE for OpenVMS VAX and OpenVMS Alpha consists of the following services:

- Remote Procedure Call (RPC) service provides connections between individual procedures in an application across heterogeneous systems in a transparent way.
- Interface Definition Language (IDL) compiler (required for developing distributed DCE applications).
- Threads service provides user-mode control and synchronization of multiple operations. Threads is packaged with the base operating system.
- Cell Directory Services (CDS) provides a location-independent method of identifying resources within a cell. A cell is the smallest group of DCE systems that share a common naming and security domain.
- Distributed Time Service (DTS) provides date and time synchronization within a cell.
- DCE Security Services provides authentication and authorization within a cell and is based upon MIT's Kerberos private key encryption system.

1.1 New Features in Version 1.3B

Version 1.3B of Digital DCE for OpenVMS VAX and OpenVMS Alpha includes the following new features. (For more information on these new features, see the *Digital DCE for OpenVMS VAX and OpenVMS Alpha Product Guide*, unless otherwise stated.)

- OpenVMS Version 7.0 Support — Digital DCE for OpenVMS V1.3B is supported on OpenVMS V7.0. (However, on Alpha systems, Digital DCE for OpenVMS only works with Kernel Threads disabled. To disable Kernel Threads, set the SYSGEN parameter MULTITHREAD to 0.)
- Integrated Login — Combines the DCE and OpenVMS login procedures.
- DCE IDL Compiler with C++ Extensions (also known as Object-Oriented RPC) — IDL has been extended to support a number of C++ language syntax features that provide a distributed object framework. The DCE RPC runtime environment now supports C++ bindings to remote objects.
- Resource Broker — The Resource Broker is Digital's Out-of-the-Box monitoring and service access tool. When deployed within a DCE cell, Resource Broker determines the services and resources in your environment, and makes them available to you through an easy-to-use graphical interface. (See the *Guide to the Resource Broker*. The Resource Broker release notes are located in SYS$HELP:RB011.RELEASE_NOTES.)
- Generic Security Service (GSSAPI) — Provides an application programming interface that lets you extend DCE security to distributed applications that handle network communications by themselves.
- Native Kerberos — Allows native Kerberos clients to use the DCE Security Server (UDP port 88) for authentication services. (See Section 13.)
- RTI (Remote Task Invocation) RPC — Provides a transactional RPC for use with Digital's ACMSxp TP product.
- RPC support for DECnet Phase V addresses — Provides RPC support for DECnet/OSI.
- XDS and X.500 support — Provides Digital X/Open XDS Directory Services and Digital X/Open Object Managment Services, and support for X.500 naming. (See the *Digital DCE for OpenVMS VAX and OpenVMS Alpha Reference Guide*.)
- A new manual, the *Digital DCE for OpenVMS VAX and OpenVMS Alpha Reference Guide* — Contains commands and reference information for Digital DCE.

2 Contents of the Kits

Digital DCE for OpenVMS has four kits available:

– Runtime Services Kit

– Application Developer's Kit:

– CDS Server Kit

– Security Server Kit

Note that the right to use the Runtime Services Kit is included as part of the OpenVMS license. The other kits each require a separate license.

The following sections list the contents of each of these kits.

2.1 Runtime Services Kit

The Runtime Services provide the basic services required for DCE applications to function. The Runtime Services Kit contains the following:

- Authenticated CDS Advertiser and Client Support
- CDS Browser
- CDS Control Program (CDSCP)
- Authenticated DCE RPC runtime support (supports DECnet, TCP, and UDP)
- Resource Broker
- RTI (Remote Task Invocation) RPC for Digital's ACMSxp TP product
- Security Client Support
- Integrated Login
- A DCE_LOGIN tool for obtaining credentials
- A RGY_EDIT tool for registry maintenance functions
- KINIT, KLIST, and KDESTROY Kerberos tools
- An ACL_EDIT tool for access control lists (ACLs) for DCE objects
- RPC Control Program (RPCCP)
- Name Services Interface Daemon (nsid); also known as the PC Nameserver Proxy
- Native Kerberos support
- XDS Directory Services
- XDS Object Managment

2.2 Application Developer's Kit

The Application Developer's Kit is used by developers to build DCE applications. The Application Developer's Kit contains the following:

- The above contents of the Runtime Services Kit
- Required DCE application development header files
- Interface Definition Language (IDL) compiler
- DCE IDL Compiler with C++ Extensions (Object-Oriented RPC)
- NIDL-to-IDL compiler (on OpenVMS VAX only)
- Generic Security Service (GSSAPI)
- LSE Templates for IDL
- UUID Generator
- .H (Include) files and .IDL files for application development
- Sample DCE applications

2.3 CDS Server Kit

The CDS Server kit provides the naming services necessary for DCE clients to locate DCE server applications. The CDS Server kit includes the following:

– CDS server (cdsd)

– Global Directory Agent (GDA)

The Global Directory Agent (GDA) lets you link multiple CDS namespaces using the Internet Domain Name System (DNS).

2.4 Security Server Kit

The Security Server kit provides security services necessary for authenticated RPC calls between DCE client and server applications to function.

– Security server (secd)

– Tool used to create the security database (sec_create_db)

– Security server administrative tool (sec_admin)

3 Installation/Configuration Prerequisites

Note that Digital DCE for OpenVMS VAX and Digital DCE for OpenVMS Alpha may require that you install a special kit before you can begin the installation of Digital DCE. See the installation and configuration guide for more information.

Make sure that you run DCE$SETUP.COM from a valid directory. Errors may occur during the installation that leave the default directory invalid.

See the first chapter in the *Digital DCE for OpenVMS VAX and OpenVMS Alpha Installation and Configuration Guide* for information on installation and configuration prerequisites.

3.1 Reconfiguring after Installation

If you are installing a new version of Digital DCE for OpenVMS VAX and OpenVMS Alpha over an existing version, you do not have to reconfigure DCE after the installation. Before the installation, stop the DCE deamons by entering one of the following commands (CLEAN is recommended):

```
$ @SYS$MANAGER:DCE$SETUP CLEAN
$ @SYS$MANAGER:DCE$SETUP STOP
```

Then, after the installation, enter the following command:

```
$ @SYS$MANAGER:DCE$SETUP START
```

If you are installing Digital DCE for OpenVMS Version 1.3B over Version 1.2, you can enable the new Integrated Login functionality from the Configuration menu without reconfiguring your host(s).

You must reconfigure if you are installing Digital DCE for OpenVMS for the first time or if you are installing a new version over Digital DCE for OpenVMS Version 1.0.

4 Troubleshooting

A chapter on troubleshooting is part of the *Digital DCE for OpenVMS VAX and OpenVMS Alpha Product Guide*. This chapter includes the following sections:

- General troubleshooting hints
- Time zone and time synchronization problems
- Client/Server Check List

5 Resource Broker

The Resource Broker is Digital's Out-of-the-Box monitoring and service access tool. When deployed within a DCE cell, Resource Broker determines the services and resources in your environment, and makes them available to you through a graphical interface. With the Resource Broker as your agent, you can find DCE servers across your cell and see their usage displayed in the form of a bar graph. This tool assists you in isolating performance bottlenecks (such as over-utilized DCE daemons or overloaded computers).

With its cell-wide view of servers and resources, the Resource Broker provides dynamic, customizable server selection algorithms that can take into account availability, appropriateness, and efficiency, and then choose to either utilize an existing server or start up a new server. The selection algorithms can utilize the attributes of the server (calls/second, packets/second, protocol sequences supported) and the attributes of the host on which the server is executing (CPU speed, CPU utilization, network usage, memory availability, and more). This dynamic server selection capability is made available in the graphical user interface, command line interface, and also transparently through the standard RPC Name Service Interface (NSI). This enables applications to replace the random-order server selection algorithm provided by the DCE standard with user-defined load balancing algorithms without modifying the application.

The release notes for the Resource Broker are located in SYS$HELP:RB011.RELEASE_NOTES.

Note: The Resource Broker runs on OpenVMS Alpha Version 6.1 or higher. It does not run on OpenVMS Alpha Version 1.5. (Resource Broker also runs on all versions of OpenVMS VAX supported by Digital DCE for OpenVMS VAX.)

6 Updates to the System Login File

To define foreign commands, have the system manager add the following to your SYLOGIN.COM after the installation:

```
$ If F$SEARCH("SYS$MANAGER:DCE$DEFINE_REQUIRED_COMMANDS.COM")-
     .NES. "" THEN @SYS$MANAGER:DCE$DEFINE_REQUIRED_COMMANDS.COM

$ If F$SEARCH("SYS$COMMON:[DCE$LIBRARY]DCE$DEFINE_OPTIONAL_COMMANDS.COM")-
     .NES. "" THEN @SYS$COMMON:[DCE$LIBRARY]DCE$DEFINE_OPTIONAL_COMMANDS.COM
```

7 Sizing for a Large Number of Users

The DCE daemons require a number of system resources for each concurrent DCE client or server process. The default number of resources allocated to the daemons is based on a maximum of 70 concurrent users (servers and clients) running on a node. If you are running more than 70 DCE users on a node, you must do the following:

1. Stop DCE if it is running.
2. Define a system-wide logical called DCE$MAX_USERS to the maximum number of users desired. For example, to configure DCE for a maximum of 80 users, enter the following:

   ```
   $ define/system dce$max_users 80
   ```

   Add this command to your system startup command file so that it is executed prior to starting DCE.
3. Restart DCE.

Refer to Section 9.1 for information about adding UCX sockets if the current number of sockets is insufficient for the number of DCE users running on the node.

8 Support for Applications

The Application Developer's Kit provides support for building DCE applications using DCE Services. It provides Application Programming Interfaces (APIs) to RPC communication services, security services, and CDS name services via the RPC Name Services Interface (NSI). (Version 1.1 of Digital DCE for OpenVMS VAX and OpenVMS Alpha replaced the Local Directory Services (LDS) with the Cell Directory Services (CDS).) The Application Developer's Kit contains the IDL compiler and Runtime support. The header files and IDL files for developing applications are installed in the following directory:

SYS$COMMON:[DCE$LIBRARY]

DCE applications must also be linked with the following shareable image:

SYS$LIBRARY:DCE$LIB_SHR.EXE

This image provides the entry points and global symbol definitions for the DCE API services.

A link options file, SYS$COMMON:[DCE$LIBRARY]DCE.OPT, is also provided. It is recommended that this options file be included when linking your DCE applications.

For example:

```
$ LINK PROG,DCE:DCE/OPT
```

Linking applications in this way makes your build procedures more portable between OpenVMS VAX and OpenVMS Alpha. It also prevents link environment changes from requiring changes to command files.

9 Using TCP/IP Services for OpenVMS (UCX) with DCE

Version 1.3B of Digital DCE for OpenVMS VAX and OpenVMS Alpha requires modification of several UCX parameters for proper operation. You should carefully look through the parameters discussed in the next sections to understand any impact they may have on your local system.

All parameter changes described below, except for the *cdsLib* service definition, involve volatile parameters. That is, if UCX is restarted on your system, the parameter settings revert back to UCX-defined defaults, unless the configuration is also modified. The appropriate commands to modify both the volatile and configuration values are shown in the following sections.

9.1 Sufficient UCX Sockets

DCE RPC and CDS use UCX sockets for interprocess communication. The UCX default maximum number of sockets is inadequate for most DCE sites. It is recommended that this parameter be set to a value of at least 250. Your site may require a higher value if you are using UCX for other than DCE. To modify the number of UCX sockets, enter the following commands with the appropriate value for the number of sockets. For example:

```
$ UCX SET COMMUNICATION /DEVICE_SOCKETS=250
$ UCX SET CONFIGURATION COMMUNICATION /DEVICE_SOCKETS=250
```

If the number of sockets is insufficient for the number of DCE users running on the node, increase the number of device sockets by two for each additional DCE user (client or server).

9.2 Sufficient UCX Small and Large Buffers

The number of UCX small and large buffers necessary for proper performance depends on the number of network software applications running on your system. As a minimum for DCE sites, the following values are recommended:

Maximum Small Buffers = 600
Maximum Large Buffers = 200

Before you configure DCE, you should check the maximum and peak values for both small and large buffers as follows:

```
$ UCX SHOW COMMUNICATION
$ UCX SHOW COMMUNICATION/MEMORY
```

A nonzero drop value or a nonzero wait value indicates that you should increase the maximum buffer value. In general, the maximum value should be at least 20 percent higher than the peak value. Additionally, these counts will change in the future, and should be checked periodically, making adjustments as necessary. For example:

```
$ UCX SET COMMUNICATION/SMALL=(MAXIMUM:600)
$ UCX SET CONFIGURATION COMMUNICATION/SMALL=(MAXIMUM:600)

$ UCX SET COMMUNICATION/LARGE=(MAXIMUM:200)
$ UCX SET CONFIGURATION COMMUNICATION/LARGE=(MAXIMUM:200)
```

See the UCX system management guide for more information on tuning UCX.

9.3 UCX TCP Protocol Settings

DCE CDS is sensitive to the values of the TCP protocol parameters of the underlying TCP communication package. Improperly setting these parameters may cause CDS client operations to appear to hang. (Hangs occur when the TCP parameters are incorrectly set and CDS client operations initiate operations that result in very large data messages being transferred between CDS clients and servers.) If this happens, other CDS clients continue to function and the hung client process may be aborted.

You can examine the current settings of the UCX TCP protocol parameters with the command:

```
$ UCX SHOW PROTOCOL TCP /PARAMETER
```

9.3.1 OpenVMS UCX TCP Parameter Settings

The correct default settings for the UCX TCP protocol parameters on OpenVMS VAX systems are as follows:

```
$ UCX SHOW PROTOCOL TCP /PARAMETERS
TCP
  MTU size segment:      disabled
  Delay ACK:             disabled
  Loopback:              disabled
  Drop timer:                 600
  Probe timer:                 75

                          Receive               Send
  Checksum:               enabled             enabled
  Push:                  disabled            disabled
  Quota:                     4096                4096
```

Note that the TCP /LOOPBACK and TCP/DELAY_ACK parameters must be disabled on Digital DCE for OpenVMS.

If either of these parameter settings do not match the default settings above, enter one of the following sets of commands:

```
$ UCX SET PROTOCOL TCP /NODELAY
$ UCX SET CONFIGURATION PROTOCOL TCP /NODELAY

$ UCX SET PROTOCOL TCP /NOLOOPBACK
$ UCX SET CONFIGURATION PROTOCOL TCP /NOLOOPBACK
```

9.4 cdsLib Service Definition

CDS uses a TCP service definition in the UCX services database. This service defines the port number for CDS client and clerk communication. The DCE$SETUP CONFIGURE operation should properly define this service for you. By default, port number 1234 is used. If your site has another application that has defined a service using port 1234, the CONFIGURE operation will ask you to choose another port number for use with the cdsLib service.

After Digital DCE for OpenVMS is configured, should you need to change the port number assigned to the cdsLib service (for example, you want to install an application that needs port 1234), use the following commands:

```
$ UCX SET NOSERVICE "cdsLib"
```

The current service definition is displayed and you are asked if you wish to delete it. Answer YES and enter the following command.

```
$ UCX SET SERVICE "cdsLib" /PORT=nnnn /file=NL: -
       /USER=DCE$SERVER /PROTOCOL=TCP /PROCESS=DCE$CDSCLERK
```

Where nnnn is an unused port number to be used by CDS.

Note that three additional ports are defined:

- cdsAdver uses port number 1235 for process DCE$CDSADV
- cdsDiag uses port number 1236 for process DCE$CDSD
- kerberos5 uses port number 88 for process DCE$SECD

```
$ UCX SHOW SERVICE
```

This command lets you examine the current UCX service definitions.

The *State* for all of the DCE services should be *Disabled*.

Also note that the service definitions in UCX are permanent settings; that is, once defined, they will still be set if UCX is restarted. For this reason, you do not need to put changes to the service definitions in your UCX startup procedure.

10 Using MultiNet with DCE

Version 1.3B of Digital DCE for OpenVMS can be used with TGV, Inc.'s MultiNet product in place of Digital's TCP/IP Services for OpenVMS. If you wish to use MultiNet with Digital DCE for OpenVMS, you must contact TGV, Inc. for a copy of MultiNet which contains support for DCE.

Then, follow the installation procedure and choose MULTINET when the installation process prompts you for the specific TCP/IP product you want to use.

Add or replace the following command to the system startup command procedure (SYS$MANAGER:SYSTARTUP.COM) after the startup commands for the network transports, DECnet and/or DEC TCP/IP Services:

```
$ @SYS$STARTUP:DCE$STARTUP START MULTINET
```

To configure DCE with MultiNet, enter the following command:

```
@SYS$STARTUP:DCE$STARTUP CONFIG MULTINET
```

Otherwise, DCE will expect TCP/IP communications to be provided by UCX.

The SYSGEN parameter MAXBUF must be set to a value greater than the maximum message size to be transferred between the CDS Clerk and CDS clients. If MAXBUF is not large enough, client processes will hang in an I/O wait state. If this happens, other CDS clients will continue to function and the hung process may be aborted without affecting them. The recommended setting for MAXBUF is 20,000 bytes or greater. (If you have a large CDS database with many directories, you may have to set it even higher.) If DCE processes hang while performing name service requests that transfer larger amounts of data, you probably need to increase the value of MAXBUF as follows:

```
$ RUN SYS$SYSTEM:SYSGEN
SYSGEN> USE ACTIVE
SYSGEN> SET MAXBUF nnnn  ! nnnn = new value for MAXBUF
SYSGEN> WRITE ACTIVE
SYSGEN> USE CURRENT
SYSGEN> SET MAXBUF nnnn  ! nnnn = new value for MAXBUF
SYSGEN> WRITE CURRENT
SYSGEN> EXIT
```

Note that this setting will remain in effect until the next time AUTOGEN is invoked. Make the changes permanent by editing SYS$SYSTEM:MODPARAMS.DAT and adding MIN_MAXBUF = nnnn and then invoking AUTOGEN as described in the installation and configuration guide.

For further information on modifying SYSGEN parameters or on AUTOGEN, refer to the OpenVMS system management documentation.

11 Using PathWay with DCE

Version 1.3B of Digital DCE for OpenVMS has been designed to be used with Wollongong's PathWay product in place of Digital's TCP/IP Services for OpenVMS.

If you wish to use PathWay with Digital DCE for OpenVMS, you must contact Wollongong for availability information and for a copy of PathWay that contains support for DCE. (Wollongong has not yet released a version of PathWay that supports Digital DCE for OpenVMS.)

Then, follow the installation procedure and choose PATHWAY when the installation process prompts you for the specific TCP/IP product you want to use.

Add the following command to the system startup command procedure (SYS$MANAGER:SYSTARTUP.COM) after the startup commands for the network transports, DECnet and/or DEC TCP/IP Services:

```
@SYS$STARTUP:DCE$STARTUP START PATHWAY
```

To configure DCE with PathWay, enter the following command:

```
@SYS$STARTUP:DCE$STARTUP CONFIG PATHWAY
```

Otherwise, DCE will expect TCP/IP communications to be provided by UCX.

12 Using TCPware with DCE

Version 1.3B of Digital DCE for OpenVMS can also be used with Process Software's TCPware product in place of Digital's TCP/IP Services for OpenVMS. If you wish to use TCPware with Digital DCE for OpenVMS, you must contact Process Software for a copy of TCPware which contains support for DCE.

Then, follow the installation procedure and choose TCPWARE when the installation process prompts you for the specific TCP/IP product you want to use.

Add the following command to the system startup command procedure (SYS$MANAGER:SYSTARTUP.COM) after the startup commands for the network transports, DECnet and/or DEC TCP/IP Services:

```
@SYS$STARTUP:DCE$STARTUP START TCPWARE
```

To configure DCE with TCPware, enter the following command:

```
@SYS$STARTUP:DCE$STARTUP CONFIG TCPWARE
```

Otherwise, DCE will expect TCP/IP communications to be provided by UCX.

13 Kerberos

The DCE Security Server makes UDP port 88 (service name "kerberos5") is available for use by native Kerberos clients for authentication.

Kerberos realm names must match the cell name of the DCE security server.

Support for native kerberos5 clients has undergone minimal interoperability testing. Interoperability has been tested between Digital DCE for OpenVMS V1.3B and MIT Kerberos V BL2. On non-OpenVMS platforms, interoperability fails when you use DCE Security with versions of MIT Kerberos V later than BL2. Full support will be provided in a future version of Digital DCE for OpenVMS VAX and OpenVMS Alpha.

14 Linking RPC Stub Modules into Shareable Images

If you build shareable images that contain RPC generated stub modules, you should use a linker options file. PSECT statements in the linker options file are used to resolve differences in the PSECT attributes between the RPC generated object file and the new shareable image. The following sections discuss how to solve problems that can arise when you create, link against, or activate a shareable image that contains RPC generated stub modules. This section can be summarized as follows:

- Program sections (PSECTs) in shareable images should be `SHR,NOWRT` or `NOSHR,WRT` unless the image is installed with priviledges.
- Program sections in modules linked against shareable images must match exactly or conflicting PSECT errors will occur.
- Until the program runs, you may have to correct PSECT attributes as far back as the shareable image.

The PSECT attributes of the RPC generated interface specifications (IFspecs) should be set to the following:

```
(GBL,SHR,NOWRT)
```

RPC interface specs usually do not change, so it is rarely required that they be set to a writable PSECT attribute. RPC interface specs are frequently shared. If your shareable image contains more than one cluster and the same interface spec is defined in multiple object modules, these interface specs can be effectively collected into the same global cluster with the GBL PSECT attribute. Note that, in this case, the first module encountered by the linker that defines the IFspec will be used to initialize the value of the IFspec in the shareable image. A map file can help you identify and correct problems with PSECTs and their contents. The contents of any PSECT should be non-zero.

If you find a zero byte PSECT, you may need to explicitly specify the module name in the options file. The module name can be specified directly on its own or as part of the `/library/include=()` statement associated with an object library. PSECTs should not be zero unless they are initialized at runtime, and this presumes that the PSECT is writable `(WRT)`.

14.1 Errors Creating a Shareable Image

The following examples show some of the errors that might occur when you try to create a shareable image with RPC stub object modules.

```
$ link/share/exe=myshr.exe/map=myshr.map -
_$ test1_mgr,test1_sstub,dce:dce.opt/opt
%LINK-I-BASDUERRS, basing image due to errors in relocatable references
%LINK-W-ADRWRTDAT, address data in shareable writeable section
in psect TEST1_V0_0_S_IFSPEC offset %X00000000
in module TEST1_SSTUB file USER:[MY.CODE.DCE]TEST1_SSTUB.OBJ;
$
```

The PSECT name is causing the linker problem. To correct this problem, create an option file including the following line, and place it on your link command line as follows:

```
$ create myopt.opt
  PSECT= TEST1_V0_0_S_IFSPEC, shr,nowrt,gbl
  ctrl-z
$
$ link/share/exe=myshr.exe/map=myshr.map -
$_ test1_mgr,test1_sstub,dce:dce.opt/opt,myopt.opt/opt
```

This will remove the link problems so that you can create a shareable image. There are still errors in this shareable image whose solutions are shown in the following examples.

14.2 Errors Linking Against a Shareable Image

Once you have a shareable image, you may still see linker problems related to the PSECT attributes between the shareable image and new object files. In the following example, a main routine is linked against the same shareable image from the previous example. The new object module references some of the same variables defined by the RPC stub module.

```
$ link/exec=test1d/map=test1d.map test1_main,sys$input/opt
  myshr.exe/share
  ctrl-z
%LINK-W-MULPSC, conflicting attributes for psect TEST1_V0_0_S_IFSPEC
in module TEST1_MAIN file USER:[MY.CODE.DCE]TEST1_MAIN.OBJ;
$
```

If you search the map files of both `myshr.map` and `test1d.map` for the PSECT TEST1_V0_0_S_IFSPEC, you will see that the PSECT attributes for this PSECT match; however, the map files are incorrect. The solution to this link problem is to include the PSECT directive in a linker options file for the offending PSECT name. The previous example simply typed in the options from the command line, but you should place these linker statements in a linker option file. The options are typed in from SYS$INPUT in the following example.

```
$ link/exec=test1d/map=test1d.map test1_main,sys$input/opt
  PSECT= TEST1_V0_0_S_IFSPEC, shr,nowrt,gbl
  myshr.exe/share
  ctrl-z
$
```

14.3 Errors Activating Shareable Images

When you run this program, the following results occur:

```
$ run test1d
%DCL-W-ACTIMAGE, error activating image MYSHR
-CLI-E-IMAGEFNF, image file not found SYS$LIBRARY:MYSHR.EXE
$
```

To allow the image activator to check a directory other than SYS$LIBRARY for your new shareable image, you must define a logical name or you must copy your new shareable image into SYS$LIBRARY. In the following example, a logical name is defined and the program is run again with the following results.

```
$ define MYSHR sys$disk:[]myshr.exe;
$
$ run test1d
%DCL-W-ACTIMAGE, error activating image MYSHR
-CLI-E-IMGNAME, image file USER:[MY.CODE.DCE]MYSHR.EXE;
-SYSTEM-F-NOTINSTALL, writable shareable images must be installed
$
```

The problem is in the `myshr.exe` image: `myshr.exe` has PSECTs whose PSECT attributes specify *both* `SHR` and `WRT`. The solution is to add the correct PSECT attributes to the offending PSECTs in the `myshr.exe` shareable image to `myshr.opt`. This can be done on the command line, as follows:

```
$ link/share/exe=myshr.exe/map=myshr.map -
$_ test1_mgr,test1_sstub,dce:dce.opt/opt,sys$input/opt
 psect= TEST1_V0_0_S_IFSPEC, shr,nowrt,gbl
 psect= RPC_SS_ALLOCATE_IS_SET_UP, noshr,wrt,gbl
 psect= RPC_SS_CONTEXT_IS_SET_UP, noshr,wrt,gbl
 psect= RPC_SS_SERVER_IS_SET_UP, noshr,wrt,gbl
 psect= RPC_SS_THREAD_SUPP_KEY, noshr,wrt,gbl
 psect= RPC_SS_CONTEXT_TABLE_MUTEX,noshr,wrt,gbl
 psect= TEST1_V0_0_C_IFSPEC, shr,nowrt,gbl
 ctrl-z
$
```

All of the PSECTs that existed in the `myshr.map` mapfile that had `SHR` and `WRT` attributes were changed so that the PSECT was either `SHR,NOWRT` or `NOSHR,WRT`. The choice depends upon your use of the data item. IFspecs are usually shared and nonwritable. The RPC_SS PSECTs are written and not generally shared among program images linked against the shareable image.

The following example tries to relink the main program again, but another problem occurs:

```
$ link/exec=test1d/map=test1d.map test1_main,sys$input/opt
  PSECT= TEST1_V0_0_S_IFSPEC, shr,nowrt,gbl
  myshr.exe/share
  ctrl-z

%LINK-W-MULPSC, conflicting attributes for psect TEST1_V0_0_C_IFSPEC
in module TEST1_MAIN file USERE:[MY.CODE.DCE]TEST1_MAIN.OBJ
$
```

Because the PSECT attributes of the TEST1_V0_0_S_IFSPEC PSECT was changed in the shareable image, its reference in `test1_main.obj` is not correct. To solve this problem, add the correct PSECT attribute. For example:

```
$ link/exec=test1d/map=test1d.map test1_main,sys$input/opt
  PSECT= TEST1_V0_0_S_IFSPEC, shr,nowrt,gbl
  PSECT= TEST1_V0_0_C_IFSPEC, shr,nowrt,gbl
  myshr.exe/share
  ctrl-z

$
```

In the final example, the `test1d` program is run and the desired results occur.

```
$ run test1d
  ncacn_ip_tcp 16.32.0.87 3314
  ncacn_dnet_nsp 63.503 RPC270002590001
  ncadg_ip_udp 16.32.0.87 1485
```

15 Support for Workstations

Digital DCE for OpenVMS VAX and OpenVMS Alpha Version 1.3B contains features that support users who access DCE services from workstations, as follows:

- DCE menu in the DECwindows session manager
- DCE$REFRESH utility that refreshes the login credentials provided by Integrated Login beyond the default DCE credential lifetime

15.1 DCE Menu in the DECwindows Session Manager

Digital DCE for OpenVMS V1.3B includes an additional menu in the DECwindows session manager. This menu makes DCE easier to use for those users accessing DCE services from a DECwindows workstation. The DCE menu, shown below, provides easy-to-invoke access to the following DCE services:

Namespace Editor
Resource Broker

Show DCE Credentials
Refresh DCE Credentials
Change DCE Password

DCE Registry Editor
CDS Control Program
RPC Control Program

15.2 DCE$REFRESH Utility to Refresh Login Credentials

Digital DCE for OpenVMS V1.3B provides the DCE$REFRESH utility that refreshes the login credentials provided by Integrated Login beyond the default DCE credential lifetime. (Login credentials expire at regular intervals unless they are refreshed.)

If you are a DECwindows workstation user, you can automatically refresh your login credentials by starting the DCE$REFRESH utility in one of the following ways:

- Select *Refresh DCE Credentials* from the DCE menu
- Run SYS$SYSTEM:DCE$REFRESH

The DCE$REFRESH utility asks for your DCE password and then runs in the background. It automatically refreshes your login credentials just before they expire.

You can interactively refresh credentials by entering the `kinit` command, but you must repeat that command each time the credentials are about to expire.

16 Restrictions and Known Bugs

The following sections provide details on restrictions and known bugs in this version of Digital DCE for OpenVMS.

16.1 OpenVMS Version 7.0 Support

Digital DCE for OpenVMS V1.3B is supported on OpenVMS V7.0. However, on Alpha systems, Digital DCE for OpenVMS only works with Kernel Threads disabled. To disable Kernel Threads, set the SYSGEN parameter MULTITHREAD to 0.

When installing Digital DCE for OpenVMS on Alpha V7.0, you may see the following error message:

```
%INSTALL-E-FAIL, failed to CREATE entry for DTSS$RUNDOWN.EXE
-INSTALL-E-SYSVERDIF, system version mismatch - please relink
```

You can ignore this error. The correct version of DTSS$RUNDOWN.EXE is installed by DCE.

16.2 Preinstallation Kit on OpenVMS Alpha

Do not be concerned if you see the following message during your installation of the DCEPRE013 kit on OpenVMS Alpha Version 1.5:

```
%INSTALL-E-RESFAIL, failed to install image with /RESIDENT qualifier
-INSTALL-E-NOGHREG, insufficient memory in the code granularity hint region
%INSTALL-I-NONRES, installed image non-resident with other specified options
```

This error occurs because the DECC$SHR.EXE image is already installed /RESIDENT and there is not enough space to install the new DECC$SHR.EXE /RESIDENT. You should reboot as soon as you can to resolve the problem. Until you reboot, you may experience a slight decrease in performance.

16.3 Minimum Global Pages

Digital DCE for OpenVMS VAX and OpenVMS Alpha Version 1.3B has increased its global pages requirements as follows:

- Digital DCE for OpenVMS VAX now requires 3750 global pages before installation. (Previously the requirement was 3000.)
- Digital DCE for OpenVMS Alpha now requires 7350 global pages before installation. (Previously the requirement was 6000.)

16.4 Upgrading from the Digital DCE Developers' Kit

Before you upgrade from the Digital DCE Developers' Kit for OpenVMS VAX Version 1.0 to Digital DCE Version 1.3B, you should shut down DCE Version 1.0 on the system. If DCE Version 1.0 is running in a cluster, be sure to repeat the shutdown procedure on all nodes of the cluster. The command to do this is:

```
$ @SYS$MANAGER:DCE$SETUP STOP
```

If you do not perform this step before installing Version 1.3B of DCE for OpenVMS, DCE$SETUP will not detect the presence of the Local Directory Service Daemon (LDSD) on your system and will not shut it down. The DCE$LDSD process will remain running on the system until the next reboot

or until it is stopped with the STOP /ID=pid command. LDS is an obsolete facility that was replaced by CDS in Version 1.1 of Digital DCE for OpenVMS.

Note that if LDSD remains active on your system, it will have NO effect on the operation of Version 1.3B of Digital DCE for OpenVMS. RPC looks only for the Cell Directory Services (CDS) facility for name service operations.

16.5 Interoperability with the DCE Developers' Kit

No specific testing has been performed with the DCE Developers' Kit for OpenVMS Version 1.0. However, the RPC and Threads components of the Developers' Kit are prerequisites and should work as expected.

16.6 Support for RPC Only Configurations

Digital DCE for OpenVMS Version 1.3B supports the use of RPC without the use of other DCE components or the need to configure a DCE cell. Because OpenVMS DCE is based on Open Software Foundation's DCE source code, this functionality could become unsupported in a future version of OSF DCE, and thus unsupported in Digital DCE for OpenVMS. Digital will make every effort to continue the support of RPC ONLY configurations; however, be advised that a possibility exists that future versions may not support RPC ONLY without configuring a DCE cell.

16.7 RTI (Remote Task Invocation) RPC

RTI RPC is a transactional RPC which is provided for use with Digital's ACMSxp TP product. RTI RPC requires OSI TP from the OSI Application Developer's Toolkit.

16.8 Format of X.500 Cell Names

X.500 cell names have the form `c=country/o=organization/ou=organization unit`. X.500 cell names can contain spaces or hyphens if they are enclosed in double quotes, but underscores are never allowed, even if they are enclosed in double quotes. For example, the X.500 cell names `/c=us/o=digital /ou="excess cell"` and `/c=us/o=digital/ou="excess-cell"` are allowed, but `/c=us/o=digital/ou=excess_cell` and `/c=us/o=digital/ou="excess_cell"` are not allowed.

16.9 Shutting Down Digital DCE for OpenVMS Before Reinstallation

If you are installing Digital DCE for OpenVMS Version 1.3B over an existing version of DCE on a common system disk in a VMScluster environment, *be sure to shut down DCE on all nodes that share the common system disk* before the installation. If you do not shut down DCE and you select the PURGE option within VMSINSTAL, parts of DCE and your OpenVMS cluster may exhibit undesirable characteristics.

If you are reinstalling Digital DCE for OpenVMS Version 1.3B over a Version 1.3 field test kit and you are using Integrated Login, and if you do not shut down DCE on all nodes that share the common system disk, you can cause the LOGINOUT image to fail to run on all of the nodes that share the common system disk.

You can correct this problem by shutting down and restarting DCE on the affected nodes. However, if LOGINOUT is not running you cannot log in; therefore you must reboot the system to correct the problem.

16.10 Configuring a CDS Replica Clearinghouse

Before you configure a CDS replica clearinghouse, make sure that the system clock is synchronized to within seconds of the CDS master server. To validate the time, use the following command:

```
$ dtscp show dce local servers
```

This shows the skew between the host and all other DTS servers in the cell.

16.11 Reconfiguring a CDS Replica Clearinghouse

If it becomes necessary to reconfigure or rebuild a host that includes a CDS replica clearinghouse, you may find that the creation of the clearinghouse succeeds but the skulk that is executed immediately after fails. If this happens, you will see the following message:

```
*** The creation of the CDS Replica Clearinghouse has succeeded
*** but the namespace has been left in an inconsistent state.
*** This condition will correct itself in a short period of time.
*** Once the command "cdscp set dir /.: to skulk" can be
*** successfully executed the namespace will be consistent and
*** the replica clearinghouse will be fully operational.
*** In the meantime you can replicate directories.
```

This is a known problem. The situation will clear itself in about an hour; however, you will not be able to create any other clearinghouses until this condition has been corrected.

If you want to correct the problem immediately, you can restart DCE on the master server. You will then be able to skulk the root directory and add additional clearinghouses.

16.12 Privileged User Refreshing Credentials

When a priviledged process creates or refreshes credentials, the owner UIC for the files is [DCE$SERVER]. If a privileged process needs to refresh credentials for an unprivileged process, the privileged process should first change its owner UIC to be the same as the unprivileged process and disable its privileges. Otherwise, the owner UIC for the updated credentials will be [DCE$SERVER], and the unprivileged process may no longer be able to read its own credentials.

16.13 Support for Integrated Login Before DCE Startup on OpenVMS Systems

If your OpenVMS system startup allows interactive logins to occur *before* DCE is started, the interactive logins that occur before DCE is started will not support Integrated Login.

If you interactively log in to OpenVMS before DCE is started, you must specify your OpenVMS username and password. You will not be logged in with DCE credentials. (If you log in after DCE is started on systems where Integrated Login is enabled, it is recommended that you specify your DCE principal name and password at the username and password prompts when using Integrated Login.)

16.14 Support for Integrated Login Before DCE Startup on OpenVMS Workstations

If your OpenVMS system startup allows DECwindows Motif to start up and display the DECwindows login box *before* DCE is fully started, the first DECwindows login will not support Integrated Login. In this case, Integrated Login will not be supported even if the first login occurs after DCE is up and running.

If DECwindows Motif displays the DECwindows login box before DCE is started, you must specify your OpenVMS username and password. You will not be logged in with DCE credentials. (If the DECwindows login box is displayed on your workstation after DCE is started and Integrated Login is enabled, it is recommended that you specify your DCE principal name and password at the username and password prompts when using Integrated Login.)

16.15 32-Character Restriction on DCE Principal Names for Integrated Login

When you log in to an OpenVMS system that has Integrated Login enabled, you can specify either your OpenVMS username or your DCE principal name at the username prompt. However, the DCE principal name you specify can contain no more than 32 characters. If your principal name and cell name combination contains more than 32 characters, specify the OpenVMS username that is associated with your DCE account instead. (This username is entered in the DCE$UAF file.) You should still enter your DCE password to obtain DCE credentials even if you specify your OpenVMS username.

16.16 Running DCE IMPORT in Batch Mode Without Password

If you run DCE IMPORT in batch mode and you do not supply a password for the DCE account on the command line, the *password valid* flag incorrectly remains set in the DCE registry. Because a password was not supplied, the flag should indicate *password not valid* and the user should not be allowed to log in. In Version 1.3B, a scan of the DCE account via RGY_EDIT reveals the incorrect flag setting (*password valid* when actually the password is not valid). However, the user will not be allowed to log in (which is the correct behavior). This problem will be corrected to indicate an invalid password in the next release.

16.17 Conversion of DCE$UAF.DAT Files Created Before Field Test Update

This note applies to field test customers who have DCE$UAF.DAT files created by the initial version of DCE field test software and who did not convert those files after installation of field test update software.

Integrated Login depends on a supplement to the system UAF file named DCE$UAF.DAT. This file is stamped with internal version information that prevents version skew between the file and the control library (API) used to manipulate the file.

For customers upgrading from Digital DCE for OpenVMS V1.3 field test to the final V1.3B software, Integrated Login cannot be enabled until the existing DCE$UAF.DAT file has been converted to a newer format. DCE$SETUP.COM checks for an out-of-rev file and prevents Integrated Login from starting if one is found, and a message describing the need for file conversion is displayed.

If Digital DCE for OpenVMS is installed in a cluster environment, you must do one of the following:

- Convert a cluster-wide DCE$UAF.DAT file and share the latest DCE software with all cluster-members.

- Split the cluster members into two sets: one with the older DCE version and one with the newer. The newer set would need to access a converted DCE$UAF file from a system-specific directory and the older set would need to access an *unconverted* DCE$UAF file from a system-specific directory. (You would forgo the capability of sharing the file until all cluster members are running the latest DCE software.)

To convert DCE$UAF.DAT to a newer format, perform the following steps:

1. Complete the installation of the final V1.3B software. If you did not see a message concerning file conversion, your file is already of the correct version (or no DCE$UAF.DAT file currently exists). You need not concern yourself further with file conversion.
2. Copy the DCE$UAF.DAT file (usually in SYS$SYSTEM) to OLD-DCE$UAF.DAT.
3. From any convenient directory, run the utility SYS$UPDATE:DCE$CONVERT_UAF_FILE. It will either convert the DCE$UAF file or print a message explaining that file conversion is not necessary. On successful conversion, the file NEW-DCE$UAF.DAT file is created in the current default directory.
4. Copy NEW-DCE$UAF.DAT file to SYS$SYSTEM:DCE$UAF.DAT.
5. Enable integrated login with DCE$SETUP.COM.

16.18 Potential Integrated Login and SYSGEN Problems

The Integrated Login component of Digital DCE for OpenVMS uses the SYSGEN parameter LGI_CALLOUTS. LGI_CALLOUTS must be set to 1 only in the ACTIVE SYSGEN parameter set when DCE is running with Integrated Login enabled. LGI_CALLOUTS must never be set to 1 in the CURRENT SYSGEN parameter set — this would prevent all logins from occurring on a subsequent reboot of the system. The following paragraphs discuss the reasons for this restriction and solutions if the problem occurs.

If Integrated Login is enabled on your system, the DCE startup and configuration procedure, DCE$SETUP.COM, sets the SYSGEN parameter LGI_CALLOUTS to 1 in the ACTIVE SYSGEN parameter set when DCE is started and resets the parameter when DCE is shut down. LGI_CALLOUTS must never be set to 1 in the CURRENT SYSGEN parameter set because, in that case, the next time the system is booted the LGI_CALLOUTS parameter is set in the ACTIVE SYSGEN parameter set *before* DCE is started. This prevents logins from occurring.

If the ACTIVE value of LGI_CALLOUTS is set to 1 when DCE and Integrated Login are not running, the following error is displayed when LOGINOUT attempts to run (for example, for interactive or batch logins):

```
No logical name match
```

Consequently, all users are prevented from logging in to the system.

This problem can occur if, for example, a SYSGEN parameter is modified in the following way while Integrated Login is enabled. This prevents logins because it causes LGI_CALLOUTS to be set to 1 the next time the system is booted.

```
$ RUN SYS$SYSTEM:SYSGEN
SYSGEN> SET param value
SYSGEN> WRITE CURRENT
SYSGEN> EXIT
$
```

The correct way to modify a SYSGEN parameter is to make the change in MODPARAMS.DAT and then run AUTOGEN. If it is essential to modify a SYSGEN parameter without using MODPARAMS.DAT and AUTOGEN, you must ensure that if you use ACTIVE, you write the parameters into ACTIVE only; and if you use CURRENT, you write the parameters into CURRENT only. Do *not* copy the ACTIVE parameters into CURRENT.

Following are two examples of acceptable ways to modify a SYSGEN parameter:

```
$ RUN SYS$SYSTEM:SYSGEN
SYSGEN> USE CURRENT
SYSGEN> SET param value
SYSGEN> WRITE CURRENT
SYSGEN> EXIT
$
```

```
$ RUN SYS$SYSTEM:SYSGEN
SYSGEN> USE ACTIVE     ! optional, default is ACTIVE
SYSGEN> SET param value
SYSGEN> WRITE ACTIVE
SYSGEN> EXIT
$
```

If you cannot log in because LGI_CALLOUTS is set to 1 and DCE is not running, there are two solutions, as follows:

— If you are already logged into the system, use SYSGEN to correct the problem.

```
$ RUN SYS$SYSTEM:SYSGEN
SYSGEN> SET LGI_CALLOUTS 0
SYSGEN> WRITE ACTIVE
SYSGEN> EXIT
$
```

— Reboot the system with a conversational boot and ensure the LGI_ CALLOUTS parameter is zero.

```
SYSBOOT> SET LGI_CALLOUTS 0
SYSBOOT> C
```

16.19 Support for Packet Privacy

Digital DCE for OpenVMS does not support the `rpc_c_prot_level_pkt_privacy` level of data encryption. However, a separate kit, the Digital DCE Privacy Option Version 1.3 for OpenVMS VAX and OpenVMS Alpha, enables an optional privacy level of encryption for RPC packets and encryption of message data when calling GSSAPI. See your Digital representative for information about ordering this kit.

16.20 DCE IDL Compiler with C++ Extensions (Object-Oriented RPC) and Upward Compatibility

Digital DCE for OpenVMS Version 1.3B includes a pre-release of the IDL compiler that is planned for the OSF DCE Version 1.2.1 release. This new version of the IDL compiler includes support for object-oriented development of C++ applications. The new support allows you to use remote objects transparently from C++.

An application using this compiler in this release will be source compatible with previous and subsequent releases; no source changes are necessary.

16.21 DCE IDL Compiler and C++ Exceptions

A client using the DCE IDL compiler with C++ extensions invokes methods on objects that causes IDL generated client stub code to be invoked. By default, communications errors or remote faults that occur during the stub's processing cause exceptions to be raised using the DCE Threads exception handling mechanism. Therefore, C++ code that needs to catch and respond to these exceptions must also use the DCE Threads exception handling mechanism.

Some, but not all, C++ compilers have built-in language support for exceptions. Currently, exceptions are not supported in the DEC C++ for OpenVMS compilers. C++ application code that processes exceptions returned from DCE IDL stubs should continue to use DCE Threads exceptions.

You can avoid the raising of exceptions from DCE IDL stubs by using the `[comm_status]` and `[fault_status]` ACF attributes. For more information, see the Guidelines for Error Handling chapter in the *DCE Application Development Guide*.

16.22 IDL Compiler Error Message when License Not Installed

The IDL compiler requires an active software license in order to execute. Since the IDL compiler is packaged in more than one software product, there are multiple licenses that it checks for when it is invoked. If it finds none of these licenses active, it issues an error message about the last license it checked for, and then exits. For example:

```
$ IDL TEST.IDL
%LICENSE-W-NOLIC, no license was found for this product - PWXXVMSDK05.01
-RMS-E-RNF, record not found
%LICENSE-F-NOAUTH, DEC PWXXVMSDK05.01 use is not authorized on this node
-LICENSE-F-NOLICENSE, no license is active for this software product
-LICENSE-I-SYSMGR, please see your system manager
```

In this case, the last license checked for was PWXXVMSDK05.01. The error message might lead you to think that the PWXXVMSDK05.01 license is required for IDL to execute on Digital DCE for OpenVMS, but it is not required. If you receive an error message similar to the one above, check to make sure that an active license (such as the Application Developer's Kit license, DCE-APP-DEV) is installed.

16.23 Automatic Registration of Servers

In the IDL compiler, servers are now automatically registered by server stubs. If you call `rpc_server_register_if()`, the "already registered" status is returned. (Remove the call to `rpc_server_register_if()` from the `server.cxx` file before you build the example programs in Chapter 19 of the *Digital DCE for OpenVMS VAX and OpenVMS Alpha Product Guide*.)

16.24 Binding to a Named Object by Object Identifier

In the IDL compiler, to bind to a named object by its object ID, the argument to the `bind()` operation is a `uuid_t` reference specifying the unique identifier of the registered object in the CDS hierarchy. Section 12.3.6.3 of the *Digital DCE for OpenVMS VAX and OpenVMS Alpha Product Guide* should specify `<interface>*bind(uuid_t &)`.

16.25 Support for sigwait()

The *DCE Application Guide* and *DCE Reference Guide* include incorrect information about support for `sigwait()`. DECthreads does not support `sigwait()` on the OpenVMS platform.

16.26 Server Programming

When running DCE server applications on OpenVMS Alpha V6.1 systems, it is possible to exhaust the server thread stack space if your server makes use of the `%f` or `%e` conversion characters for formatting output. For example, the following `printf` statement could cause an overflow of the server thread stack:

```
printf ("The computed value = %f\n", value);
```

This error can cause the server to terminate with an unexpected error code such as an Access Violation or a Reserved Operand Fault.

If you experience this type of error, you must add a call to the RPC routine `rpc_mgmt_set_server_stack_size` specifying a stack size of at least 14000, prior to calling `rpc_server_listen`.

16.27 Compiling Stubs on Alpha

If a stub is compiled on Alpha with optimization switched on, it will not handle exceptions correctly. Therefore, on Alpha, you should compile stubs with optimization switched off.

This problem will be fixed in a future release of DEC C.

16.28 Using the -cpp_cmd (/PREPROCESS) IDL Compiler Option on OpenVMS Alpha

When you specify the `-cpp_cmd` (/PREPROCESS) option in an IDL command, the IDL compiler preprocesses any IDL or ACF sources by invoking the DEC C compiler with the /PREPROCESS_ONLY qualifier. Because of a bug in the DEC C compiler on OpenVMS Alpha, the IDL compiler may incorrectly report source line numbers and contents when it reports error messages.

If your IDL and ACF source files do not use C preprocessor directives (such as #define), then you do not need to specify the `-cpp_cmd` (/PREPROCESS) option. Otherwise, the workaround is to change multiline comments to a series of single line comments.

16.29 POSIX and OpenVMS Alpha

On OpenVMS Alpha systems running POSIX Version 1.0, you should use only one page file. If there are multiple page files installed on the system, DCE_LOGIN may cause a system crash.

16.30 DEC C Applications Using G-float in the POSIX Environment

If you compile your DEC C DCE application in the POSIX environment on an OpenVMS VAX system, the DEC C compiler will not build with G-float if you specify the /G-float qualifier before the /LIBRARY qualifier on the command line.

In the following example, the DEC C compiler uses D-float even though G-float is specified:

```
$ cc/decc/list/mach/g_float x.c+DECC$LIBRARY_INCLUDE:DECC$RTLDEF.TLB/library
```

The problem is caused by the +DECC$LIBRARY_INCLUDE:DECC$RTLDEF.TLB /library part of the command. This causes the compile to reset to the default double precision type.

To work around this problem, compile your program *outside of the POSIX environment* and specify the /G_float qualifier *after* the text library, as follows:

```
$ CC/DECC -
_$ /PREFIX=(ALL,RTL="posixc$") -
_$ /INCLUDE=POSIX$INCLUDE -
_$ x.C + POSIX$INCLUDE:POSIX$CDEF/LIBRARY -
_$ /G_FLOAT
```

Refer to the *Guide to Programming with OpenVMS POSIX* for more information about compiling outside of the POSIX environment.

16.31 Information on the DCE Example Applications

Ten DCE examples are on the Digital DCE for OpenVMS kit. This section lists current restrictions:

- On OpenVMS VAX, you need the OpenVMS POSIX Version 1.3 MUP release or later in order to build and execute the Example Programs.
- The EXAMPLE_BLD.COM and makefiles, which build the example programs, select the OpenVMS VAX and Alpha `decc/vaxc` switch(es) and option files.
- The credentials cache logical KRB5CCNAME (LNM$JOB) will be accessible to POSIX processes/children only and not to subsequent psx> DCL invoked programs.
- Because FORTRAN is unavailable in the OpenVMS Alpha POSIX environment, you cannot use the Payroll example on OpenVMS Alpha at this time.
- The DCEsx example is not supported in the POSIX environment at this time.

16.32 UCX Runtime Calls Not Thread Safe

Note that UCX Runtime Calls are not always thread safe.

UCX has two main application programming interfaces: VMS system services (for example, $ASSIGN, $QIO, $CANCEL) and the C socket library. Of these two, the VMS system services are fully thread-safe, while the socket library is not. The most common problem with sockets is the `select()` call, which blocks the entire process (not just the calling thread) until the specified I/O events occur (or the timeout expires).

16.33 C RTL Routine Sleep Not Thread Safe

The C RTL routine sleep is not thread safe. The sleep call may wake up prematurely if calls to DCE APIs are made at the same time. It is recommended that you use a thread safe mechanism such as `pthread_delay_np`, `pthread_cond_wait`, `pthread_cond_timedwait`, and `pthread_cond_signal` to delay a thread. For more information on these APIs, please refer to the *OSF DCE Application Development Reference Manual*.

16.34 DECrpc Version 1.0 Compatibility

DECrpc Version 1.0 compatibility is supported only on OpenVMS VAX. It is not supported on OpenVMS Alpha.

16.35 Rebuilding DECrpc Applications Against Digital DCE

If you take advantage of the compatibility capabilities of DECrpc on Digital DCE for OpenVMS VAX, you may encounter undefined symbols when you link your application object code against the DCE library (DCE$LIB_SHR.EXE). For example, when you link the client object code of your DECrpc application interface (MY_APP) against the DCE library, you receive the following undefined symbol:

```
MY_APP_$CLIENT_EPV
```

When you link the server object code of MY_APP against the DCE library, you receive the following undefined symbols:

```
MY_APP_$MANAGER_EPV
MY_APP_$SERVER_EPV
```

The following table lists the undefined symbols you receive if you use the following DECrpc include files from SYS$COMMON:[RPC$INCLUDE] in your application when you link your application object code against the DCE library (DCE$LIB_SHR.EXE).

If your application includes...	You receive the following undefined symbols:
LLB.H	LLB_$CLIENT_EPV LLB_$MANAGER_EPV LLB_$SERVER_EPV
GLB.H	GLB_$CLIENT_EPV GLB_$MANAGER_EPV GLB_$SERVER_EPV
RRPC.H	RRPC_$CLIENT_EPV RRPC_$MANAGER_EPV RRPC_$SERVER_EPV

You can ignore these undefined symbols. They are not referenced by DCE or by application code and do not affect the running of your application in any way.

16.36 Ordering of Startup Procedures

On OpenVMS VAX and OpenVMS Alpha Version 6.1 and later, the order of startup procedures follows: DECnet, TCP/IP software, then DCE.

On OpenVMS VAX Version 5.5-2 and 6.0, the order of startup procedures follows: CRT$STARTUP.COM (the CRTL Startup Procedure), then DECnet, TCP/IP software, and finally DCE.

16.37 Case-Sensitivity of DCE Utilities

Some input to Digital DCE for OpenVMS utilities is case-sensitive (for example, CDSCP entity attribute names). Since the DCL command line interface converts all input to uppercase before passing it to a utility, some input to the DCE utilities will need to be enclosed in quotation marks (" ").

When you enter commands directly at DCE utility prompts, you should not use the quotation marks because case-sensitivity is preserved. (Case-sensitivity is not preserved by the Integrated Login utilities DCE$UAF, IMPORT, and EXPORT because these are true native OpenVMS applications.)

16.38 CDSCP Commands Requiring a Local Server

There are several CDSCP commands which assume the presence of a CDS server on the local system. These commands will not execute properly in the absence of a local server. At present, CDSCP will return the following error:

```
Failure in routine: cp-xxxxxxx
not registered in endpoint map (dce/rpc)
```

The affected commands are:

CDSCP SHOW SERVER
CDSCP DISABLE SERVER
CDSCP CREATE CLEARINGHOUSE

16.39 CDSCP DUMP CLERK CACHE Command

The CDSCP command to examine the CDS cache will fail if CDSCP is run under a Process UIC other than [DCE$SERVER].

```
$ CDSCP DUMP CLERK CACHE
Cannot map to memory cache -1
Failure in routine: cadump
Internal: Not found
```

To work around this restriction, issue the following DCL command before you invoke CDSCP:

```
$ SET UIC [DCE$SERVER]
```

Remember to reset your UIC to its original value after you use this command.

16.40 CDS Clerk Failing on UCX Shutdown

If you issue a SYS$STARTUP:UCX$SHUTDOWN command while running DCE, you may get a CDS Clerk failure and an Access Violation. You may then encounter problems restarting the CDS Clerk (and DCE itself) with the DCE$SETUP START command.

The primary problem is that UCX is being shut down while DCE is still active. Since DCE uses UCX, DCE should always be shut down first.

To recover from this problem, you need to shut down DCE first and then restart. Simply trying to restart without first shutting DCE down will not fix the underlying problem. Because temporary files may be left in an indeterminate state, you may also want to perform a DCE$SETUP CLEAN operation before restarting.

16.41 Global Directory Agent Configuration

The Global Directory Agent (GDA) is configured on the OpenVMS node that contains the CDS Master Replica name server. Make sure you choose the CUSTOM configuration option when you do the configuration. The DNS domain name (for example, zko.dec.com) and the Internet Address of an authoritative DNS Master Bind Server (for example, 16.32.2.11) are required during configuration if you are using DNS Bind style cellnames.

Before access to multiple CDS namespaces is possible, the following are required after the configuration:

1. The Master Bind Server identified during configuration becomes the repository for information the GDA requires to resolve the internet addresses and binding information needed by CDS to access foreign cell name spaces. This applies to DNS Bind cellnames only. See the Intercell Naming chapter in the *Digital DCE for OpenVMS VAX and OpenVMS Alpha Product Guide* for the binding information content, location, and access.
2. Authenticated access to foreign (intercell) cell name space requires performing the RGY_EDIT cell command. The information needed for the cell command requires coordination with the foreign cell administrator. For more information, see both the Administering a Multicell Environment chapter in the *OSF DCE Administration Guide* and the Intercell Naming chapter in the *Digital DCE for OpenVMS VAX and OpenVMS Alpha Product Guide*.
3. Before doing the RGY_EDIT cell command, you must first delete the krbtkt account for the foreign cell if one already exists. Similarly, the administrator for the foreign cell must also delete the krbtkt account in the foreign cell's registry for your cell. For example, if your cell is called first_cell and the foreign cell is called second_cell, then you must run RGY_EDIT on first_cell to delete the account called krbtkt/second_cell, and the administrator on second_cell must delete the registry account called krbtkt/first_cell.

   After the cell command, both cell administrators should rerun DCE_LOGIN before attempting authenticated cross-cell requests.

If you are unsuccessful in configuring intercell communication, check for the following:

- The clocks on the systems that are attempting to communicate show times that differ by no more than five minutes. (Use DTS to change the system time once you are running DCE.)
- CDS has the information that should be contained in the CDS_GDAPointers field in the cell's root directory. If CDS does not have this information in the cell's root directory, restart the GDA daemon process (DCE$GDAD) by entering the following commands:

  ```
  $ STOP/ID=xxxxxxxx
  $ @sys$manager:dce$setup start
  ```

  where xxxxxxxx is the PID of the DCE$GDAD process.