ENTEGRITY *Solutions*®

PC-DCE™

Administrator's Guide

Software Version 5.0

Notices

PC-DCE Administrator's Guide - Software Version 5.0- Revised March 2003

THIS DOCUMENT AND THE SOFTWARE DESCRIBED HEREIN ARE FURNISHED UNDER A LICENSE, AND MAY BE USED ONLY IN ACCORDANCE WITH THE TERMS AND CONDITIONS OF SUCH LICENSE AND WITH THE INCLUSION OF THE COPYRIGHT NOTICE BELOW. TITLE TO AND OWNERSHIP OF THE DOCUMENT AND SOFTWARE REMAIN AT ALL TIMES WITH ENTEGRITY SOLUTIONS CORPORATION OR ITS LICENSOR.

The information contained in this document is subject to change without notice.

ENTEGRITY SOLUTIONS MAKES NO WARRANTY OF ANY KIND WITH REGARD TO THIS MATERIAL DOCUMENTATION OR SOFTWARE, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Entegrity Solutions shall not be liable for errors contained herein, or for any direct or indirect, incidental, special or consequential damages in connection with the furnishing, performance, or use of this material.

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

Entegrity, Entegrity Solutions, Gradient and NetCrusader are registered trademarks or trademarks of Entegrity Solutions Corporation.

Entrust is a registered trademark of Entrust Technologies Limited. All Entrust product names are trademarks of Entrust Technologies Limited. Inprise and VisiBroker are trademarks of Inprise Corporation. Orbix is a registered trademark, and IONA and Wonderwall are trademarks of IONA Technologies. Kerberos is a trademark of Massachusetts Institute of Technology. Microsoft, Windows, and Windows NT are registered trademarks of Microsoft Corporation. Netscape and Navigator are trademarks of Netscape Communications Corporation. The Open Group is a trademark of The Open Group. VeriSign is a trademark of VeriSign, Inc. UNIX is a registered trademark in the United States and other countries, licensed exclusively through X/Open Company, Ltd. Other products and company names mentioned in the document are trademarks or registered trademarks of their respective owners.

Portions of this documentation were derived from materials provided by Entrust Technologies Limited.

Copyright © 1995–2003 The Open Group

Portions of this documentation were derived from materials provided by Compaq Computer Corporation.

Copyright © 1998 - 2003 Compaq Computer Corporation.

Permission to use, copy, modify, and distribute this software and its documentation for any purpose and without fee has been granted to Entegrity SolutionsCorporation provided that the above copyright notice and this permission notice are prominently displayed in all copies of the software and documentation, and that the name of The Open Group not be used in advertising or publicity pertaining to distribution of the software without specific, prior written permission.

THIS SOFTWARE IS PROVIDED "AS IS." THE OPEN GROUP DISCLAIMS ALL WARRANTIES, WHETHER EXPRESS OR IMPLIED, WITH RESPECT TO THIS SOFTWARE INCLUDING, WITHOUT LIMITATION, ANY WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL THE OPEN GROUP BE LIABLE FOR ANY SPECIAL, INDIRECT, OR CONSEQUENTIAL DAMAGES RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN CONTRACT, TORT INCLUDING NEGLIGENCE, OR OTHER LEGAL THEORY ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Copyright © 1998 - 2003 Entegrity Solutions Corporation & its subsidiaries. All Rights Reserved.

Entegrity Solutions Corporation, 410 Amherst St., Suite 150, Nahua, NH 03063 USA

Preface

Intended Audience

This guide is intended for administrators who want to manage their cell environment using the Entegrity® DCE implementation for Windows® called PC-DCE™ . This guide assumes a general knowledge of DCE (Distributed Computing Environment).

Documentation

This section describes the documentation that Entegrity provides with PC-DCE on both the product CD and on the Entegrity web site (**www.entegrity.com**) under the Support link:

- *PC-DCE Documentation Set*
- *The Open Group Documentation*

Documentation on other Entegrity products, such as NetCrusader/Web, is also available on the Entegrity web site.

We are always trying to improve our documentation. If you notice any inaccuracies or cannot find information, please send email to **docs@entegrity.com**. We welcome any comments or suggestions.

PC-DCE Documentation Set

The following documents are provided with PC-DCE:

- *PC-DCE Installation and Release Notes*
- *PC-DCE Overview Guide*
- *PC-DCE Administrator's Guide* (this book)
- *PC-DCE Developer's Notes*
- *PC-DCE Guide to CAS*

PC-DCE also provides online help with the following programs:

- PC-DCE Service Panel
- PC-DCE Configuration Panel
- DCE Director
- Visual DCE ACL Editor
- DCEsetup

Entegrity also provides OSF DCE Version 1.2.2 documentation on the product CD and the Entegrity Support web site.

The Open Group Documentation

The PC-DCE product CD and the Entegrity Support web site also provide The Open Group (formerly OSF) DCE Version 1.2.2 documentation, including the following guides:

- *OSF DCE Administration Guide — Core Components*
- *OSF DCE Administration Guide — Introduction*
- *OSF DCE Application Development Guide — Core Components*
- *OSF DCE Application Development Guide — Directory Services*
- *OSF DCE Application Development Guide — Introduction and Style Guide*
- *OSF DCE Application Development Reference*
- *OSF DCE Command Reference*
- *Introduction to OSF DCE*
- *OSF DCE Problem Determination Guide*
- *OSF DCE/File-Access Administration Guide and Reference*
- *OSF DCE/File-Access Users’ Guide*
- *OSF DFS Administration Guide and Reference*
- *OSF GDS Administration Guide and Reference*

Contacting Entegrity Solutions

For a complete listing of Entegrity Solutions Corporation sales, research and development, and solutions centers worldwide, please see the Entegrity web site at **http://www.entegrity.com**.

Contact	Address	Phone/Fax/Email
DCE Product and Sales Information	Entegrity Solutions Corporation 410 Amherst Street, Suite 150 Nashua, NH 03063 USA	Email: DCESales@entegrity.com Web: www.entegrity.com Tel: +1-603-882-1306 ext.2700 Toll Free (US): 1-800-525-4343 ext. 2700 Fax: +1-603-882-6092
All Other Product and Sales Information Requests	Entegrity Solutions Corporation 2001 Gateway Place, Suite 420W San Jose, CA 95110 USA	Email: info@entegrity.com Web: www.entegrity.com Tel: +1-408-487-8600 ext. 123 Fax: +1-408-487-8610
Technical Support	Entegrity Solutions Corporation 410 Amherst Street, Suite 150 Nashua, NH 03063 USA	Email: support@entegrity.com Web: support.entegrity.com Tel: +1-603-882-1306 ext. 2702 Toll Free (US): 1-888-368-3555 ext. 2702 Fax: +1-603-882-6092
Documentation Comments and Suggestions		Email: docs@entegrity.com

Contact	Address	Phone/Fax/Email
Other Inquiries	Entegrity Solutions Corporation 2001 Gateway Place, Suite 420W San Jose, CA 95110 USA	Email: info@entegrity.com Web: www.entegrity.com Tel: +1-408-487-8600 Fax: +1-408-487-8610

Obtaining Technical Support

If you purchased your DCE product directly from Entegrity Solutions or Gradient Technologies. you are entitled to 30 days of limited technical support beginning on the day the product is expected to arrive.

You may also purchase a support plan that entitles you to additional services. You *must* register prior to receiving this support. For details, refer to the customer support information package that accompanied your shipment or refer to **http://support.entegrity.com**. The web site also contains online forms for easy registration.

If you purchased PC-DCE from a reseller, please contact the reseller for information on obtaining technical support.

Obtaining Additional Technical Information

Contact	Address	Phone/Fax/Email
The Open Group™ Developer of DCE (Distributed Computing Architecture) software and standards.	The Open Group 29B Montvale Ave Woburn MA 01801 U. S. A.	Tel: +1 781-376-8200 Fax: +1 781-376-935811 **http://www.opengroup.org**

Contents

Chapter 3 Managing Users 27

Chapter 4 Authentication 47

CHAPTER 1

Planning Cell Topology

This chapter describes DCE cell topology issues and methods. It describes what cell topology is, how you can approach the task of designing a cell topology, and some of the important issues you will need to consider as you create and test your design.

If you do not yet know what a cell is, you should read a good introduction to DCE before continuing. Refer to the Preface for references.

However, to recap, a *cell* is a collection of users, machines, and resources that share two common databases of administrative information, namely, the Cell Directory Services (CDS) namespace and the security registry. Members of a cell are usually located in a common geographic area, but they can also be geographically dispersed. A cell's size can range from only one machine to several thousand.

1.1 What Is Cell Topology?

Cell topology describes how you logically and physically layer DCE onto your network of users and computing resources. Cell topology includes the division of your organization into one cell or multiple cells, and the physical and logical arrangement of DCE services within each cell.

A cell topology design includes answers to the following questions. The rest of this chapter discusses how you can approach each of them.

- How are your users distributed, functionally and geographically?
- Will you use full client software or lightweight client software?
- What are the requirements of your applications?
- Will you create one cell or multiple cells?
- If you create multiple cells, which users and resources will you assign to which cell?

- If you create multiple cells, will you allow intercell communication?
- Within a cell, where will you locate the required master DCE servers?
- How many CDS and security service replicas will you create and where will you locate them?
- What will your network and hardware requirements be, especially with respect to your servers?

1.2 Cell Topology Design and Test Methodology

The factors that influence the answers to these questions tend to be interdependent. You must consider each question separately, but you must also consider how the design issues interact. It's best to get a general feel for all of the issues involved before making specific design decisions.

A cell topology can involve complex interactions with results that are often difficult to foresee. Before making full-scale deployments, DCE cell topology designers often create test cell topologies that model the deployment environment. This allows them to test their design before committing it to their users. We recommend that you consider this approach.

You may also wish to hire consultants with experience in designing DCE cell topologies. Entegrity® provides this consulting service – for more information, contact your Entegrity sales representative.

1.3 Number and Location of Cells

If your organization is not too large, creating a single cell is the simplest solution from an administrative perspective. However, for larger organizations, you will need to consider creating multiple cells rather than a single large cell. The rest of this section discusses factors that you should consider when making this decision.

1.3.1 Performance

Multiple cells will generally provide better performance than a single large cell. In a single large cell, more users per server slows down response times.

You can improve response somewhat by strategically locating CDS and security replicas. However, users running full clients (see *Section 1.4 on page 16*) still need at startup to contact the master CDS **hosts** directory replica, which resides on a single machine. Also, there is a practical limitation on the number of replicas you can implement in a cell. Generally, more than ten replicas imposes an unreasonable administrative burden.

1.3.2 Administration

As a cell grows large it becomes increasingly difficult to administer. The sheer number of users, groups, machines, and other resources and objects becomes difficult to manage. Geographical dispersion makes the job more difficult, as does the increased number of replicas required.

Implementing multiple cells lets you divide the administration task into manageable units and delegate administrative responsibility for each cell to individual cell administrators. A multiple cell topology does have the drawback of creating some intercell administration issues. See *Chapter 7 on page 103*.

1.3.3 Geographical and Functional Distribution of Users

As you design your cell topology, consider your user population and how they are distributed, both geographically and functionally.

Are your users all located in a common geographic area, or are they located in different buildings, different cities, or different countries?

Are the members of functional groups (departments or workgroups, for example) located together, or are they geographically dispersed?

The answers to these questions affect cell topology design. If all potential users are located within the confines of a single LAN, the network performance favors a single cell topology. However, if the users are geographically dispersed, you must consider different approaches. You may choose to implement separate cells for each geographic region, or to implement a single cell with centrally-located master replicas and read-only CDS replicas and slave security replicas located at remote sites.

Because functional groups tend to share the same resources, you must consider functional distribution as well. If functional groups are organized geographically (for example, Product A’s division is located in England and Product B’s division is located in the United States), this favors multiple cells, one per group. However, if functional groups are geographically dispersed, you may consider either creating a single large cell; or multiple, geographically dispersed cells that are organized by function.

1.3.4 Cell Tuning

If you choose to implement one or more large cells, Entegrity provides cell tuning features to improve performance. These are implemented as Windows 2000, Windows NT or Windows 98 registry keys and environment variables. For example, cell tuning features allow you to increase the amount of time the DCE startup service waits for daemons to start and reduce the timeout of RPC calls to CDS and security servers.

For more information about cell tuning, refer to *Appendix A on page 117*.

1.3.5 Maintaining System Time

The systems in each cell must have a way to synchronize their local clocks with a central time server. DCE security requires system clocks to be accurate within five minutes.

PC-DCE™ provides the Distributed Time Service daemon (**dtsd**) as an option for time synchronization; however, other services may better suit your environment.

Alternatives to **dtsd** include:

- The time service provided by your network environment (NetWare, Microsoft Domain), or Network Time Protocol (NTP).
- **Timesync.exe**, which is included by default with PC-DCE. If your environment allows, running only **timesyc.exe** is a lightweight option to running **dtsd**. For details, refer to *Section 6.1 on page 91*.

1.4 Full Clients and Lightweight Clients

The Entegrity PC-DCE client can be configured either as a *full client* or a *lightweight client*. The choice to implement full or lightweight clients is a cell topology issue because lightweight clients are much more conservative of cell resources, effectively raising the practical maximum cell size and increasing the number of users that can rely on a single replica. If you implement lightweight clients, you may be able to run a single large cell in situations where you might otherwise need multiple cells.

For a complete discussion of full and lightweight clients, refer to the *PC-DCE Overview Guide*.

1.5 Application Requirements

Consider the requirements of your applications when you consider your cell design. Different applications have different performance requirements and require different types of support from DCE. For example, an application may or may not require the services of a full client, which in turn affects the practical maximum cell size (see *Section 1.4 on page 16*).

1.6 Server Locations

As part of your topology design, decide where to locate your master CDS replicas and master Security servers. Take into account performance, security, and failover considerations (see *Section 5.2 on page 66*).

1.7 Machine and Network Requirements

Your primary servers and immediate backup servers should run on high-quality hardware with adequate resources (virtual memory and physical memory) to support anticipated growth for a reasonable period of time.

1.7.1 Security Service Servers

The node that runs the master security server must be highly available and physically secure. If the host that contains the master security server goes down, slave replicas can still provide registry information, so consider having a number of replicas in each cell. Use factors such as the number of machines in your cell, the reliability of the machines that run security servers, and your cell's available resources to determine how many security replicas you need.

1.7.2 CDS Servers

CDS servers need to be located on dependable nodes.

When deciding how many CDS servers you need, consider the size of your cell and how geographically dispersed the cell is. You should have at least two copies (one master and one replica) of each CDS directory to ensure access to data if one of the servers becomes unavailable.

Use reliable network connections that can handle the estimated amount of traffic. This helps to ensure that all servers maintaining directory replicas can be reached when the CDS performs periodic updates.

1.7.3 Network Connections

If users will need to go off the local LAN to contact a server, pay careful attention that the LAN interconnections have adequate reliability and bandwidth. WAN connections are usually the source of bottlenecks.

1.8 Intercell Communication

If you implement multiple cells, you can choose to isolate them from each other, or more commonly, allow some degree of intercell communication. For example, you may want to create special foreign_user accounts so that high level DCE administrators can access multiple cells.

CHAPTER 2

Deploying in Large-Scale Environments

This chapter provides information about deploying PC-DCE in a large-scale enviroment and includes the following sections:

2.1 General Strategies

Consider employing the following strategies for expediting and simplifying large-scale client installations and upgrades:

- Keep the upgrade manageable by staggering it. Perform a limited number of upgrades at a time so that you can validate a group of systems before upgrading the next group.
- Use an automated distribution tool (*Section 2.2.2*).
- Perform manual installations of lightweight clients, rather than performing the PC-DCE installation procedure on each system (*Section 2.2.3*).

2.1.1 Configuration Tools

Either DCEsetup or PC-DCE Configuration Panel can be used to configure DCE cells. Refer to their respective online help systems for information on using either of these tools.

2.2 Installing PC-DCE on Multiple Systems

While installing PC-DCE on a single system is a straightforward task, upgrading in a large-scale environment can present challenges. You may need to configure hundreds of PC-DCE client systems. You may be deploying PC-DCE as part of an upgrade that includes multiple software products. Or, you may want to limit the installation tasks that users are asked to perform.

This section describes:

Once installation is complete, you must then configure PC-DCE clients into a cell (see *Section 2.3 on page 23*).

2.2.1 Controlling Bootup After Installation

By default, the installation program prompts the user to reboot the machine after installation. If you are using scripts to deploy software and wish to modify this default behavior, you can use the following options when running **setup**:

`C:\>` **`setup autoboot`** — The installation program automatically reboots the machine. It does not prompt the user.

`C:\>` **`setup noboot`** — The installation program does not reboot the machine and does not prompt the user.

2.2.2 Automated Distribution Tools

When PC-DCE client software needs to be installed on large numbers of systems, you can take advantage of tools that automate software distribution, such as Seagate Technologies Wininstall software. This option works for both full and lightweight client deployment.

This program takes a snapshot of a system on which PC-DCE has been installed, and packages it for distribution. When a user boots up and maps to the package's location, the model PC's configuration is duplicated on the user's system.

After installation is complete, configure each PC-DCE client into the appropriate cell (see *Section 2.3 on page 23*).

2.2.3 Manual Installation

When you need to deploy lightweight client configurations on large numbers of systems, you can write a program or use a utility that performs a manual installation of PC-DCE on many systems at once. This requires an initial investment to create the installation program, but ultimately saves time, because you do not have to install on each desktop individually.

NOTE: Because performing a manual install involves making entries in the Windows registry, you must have administrator rights to the Windows systems involved.

There are some differences in the available options between Windows 2000, Windows NT, and Windows 98:

- Windows 2000 and Windows NT allow more control over freeing the PC-DCE-related DLLs, which means a tool like Microsoft Systems Management Server (SMS) can simply stop PC-DCE, move the new binaries into the **\pcdce32\bin** directory, and start DCE again.

- With Windows 98, stopping PC-DCE does not free all the PC-DCE-related DLLs. For that reason you have to run **setup.exe** to upgrade PC-DCE. **setup.exe** can allow the install to go unattended, meaning if you already have PC-DCE installed, it will not prompt for any input and does not require a reboot on completion.

To manually install PC-DCE, first complete the standard PC-DCE lightweight client configuration on one system. Refer to the *PC-DCE Installation and Release Notes* for instructions. Use this system's configuration as the model for the other systems you are deploying.

Use the following instructions to create a program that installs PC-DCE on each system:

1. Create a top-level PCDCE32 directory on the system.
2. Copy all PC-DCE files from the model system into the target system's PCDCE32 directory, maintaining the directory structure (**bin**, **opt**, **krb5**, etc.). The **Samples** and **Docs** directories may not be needed on all systems.
3. Copy the file **dce32_ctl.cpl** into the directory:
   - For Windows 2000 and Windows NT — *operating_system***\system32** or *operating_system***\system root**
   - For Windows 98 — **Windows\system**

   dce32_ctl.cpl is the PC-DCE icon that appears in the Control Panel.
4. *If you are running a version of PC-DCE prior to 2.2,* create a **PCDCE32\licenses** directory. Copy the license file (**rtNT.lic** or **rt95.lic**) from the model system into this directory on the target system.
5. Modify the PATH environment variable to include the **PCDCE32\bin** directory.
6. Add the following registry keys and values into the Windows registry key HKEY_LOCAL_MACHINE\Software\Entegrity\DCE\Configuration:

   BoundDCERefcnt — Assign a value of 0 and a data type of REG_DWORD:

   ```
   BoundDCERefcnt:REG_DWORD:0
   ```

   DCEDir — Enter the top level PCDCE32 directory and assign a data type of REG_SZ:

   ```
   DCEDir:REG_SZ:c:\PCDCE32\
   ```

   DCEInstallType — Assign a value of 0x0000000 and a data type of REG_DWORD:

   ```
   DCEInstallType:REG_DWORD:0x0
   ```

   InstallType — Assign a value of 0x0000001 and a data type of REG_DWORD:

   ```
   InstallType:REG_DWORD:0x1
   ```

7 Add these Entegrity DCE services using the win32 API call **CreateService**. Use the following parameters:

- Display string — PC-DCE for Windows
- Service Name —DCE (might be DCE or NetCrusader/DCE)
- Binary — *DCE top level directory***bin\dce_service.exe**

Also set the following flags:

- SERVICE_ALL_ACCESS
- SERVICE_WIN32_OWN_PROCESS
- SERVICE_AUTO_START
- SERVICE_ERROR_NORMAL

8 If the system is running Windows 2000 or Windows NT, you must go into the registry and configure a dependency on RPCSS for the DCE service. Under HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services \Entegrity DCE, add the **DependOnService** value REG_MULTI_SZ:RPCSS.

NOTE: Because this registry entry is triggered by a periodic system process, there may be a delay before it is visible.

9 If integrated login is required, the network provider interface needs to be set up in the registry as follows:

- Add **ntdcelgn** to the end of the entry HKEY_LOCAL_MACHINE\ SYSTEM\CurrentControlSet\Control\NetworkProvider\Order\Provider Order. For example:

```
ProviderOrder:REG_SZ:LanmanWorkstation,ntdcelgn
```

- Add the key **ntdcelgn** to HKEY_LOCAL_MACHINE\ SYSTEM\ CurrentControlSet\Services.

  To the **ntdcelgn** key you just created, add the key **NetworkProvider**.

  To the **NetworkProvider** key you just created, add the following values and data:

  Class — Assign a value of 0x00000002 and a data type of REG_DWORD:

```
Class:REG_DWORD:0x2
```

  Name — Assign a value of PC-DCE/32 Integrated Login and a data type of REG_SZ:

```
Name:REG_SZ:PC-DCE/32 Integrated Login
```

  ProviderPath — Assign a value of *DCE top level directory***bin\ntdcelgn.dll** and a data type of REG_SZ:

```
ProviderPath:REG_SZ:d:\PCDCE32\bin\ntdcelg.dll
```

After installation is complete, reboot the system and configure it into the appropriate cell.

2.3 Configuring PC-DCE on Multiple Systems

After you install PC-DCE on all required systems, you must configure each client and add it to the appropriate cell. You can use one of the tools described in this section, or you can write your own program (such as a TCL script) to perform the tasks involved in configuration, such as contacting the CDS server and copying entries to the system.

2.3.1 How to Designate a Local Configuration Administrator

NOTE: Also called Split Configuration. Other Entegrity products, DCE for Linux and DCE for Tru64 UNIX use a similar term for something different. There, split server configuration is where the CDS and Security master servers are on different hosts in a cell.

preconfig.tcl allows local administrators to configure a full client without needing the cell administrator password. It also preconfigures each host to recognize the host-configuration permissions of these local administrators. To set up local configuration, the *cell adminstrator* follows this procedure:

(Numbered steps below correspond to the installation script prompts.)

1 Log into DCE as cell administrator.

2 Type dcecp. At the dcecp prompt, enter:

```
dcecp> source path_to_Pcdce32/opt/dcelocal/dcecp/preconfig.tcl
```

3 At “Enter the fully qualified domain names of the hosts you want to preconfigure:”

The host names must be separated by spaces only. In the example below, the cell administrator is preconfiguring foo.bar.com and my.machine.edu.

```
foo.bar.com my.machine.edu
```

4 Enter the name of the entity to which you want to grant host-configuration permission. This can be the name of a group or a principal.

In this example, the entity name is local_admin.

```
local_admin
```

5 Distinguish if this entity is a group (g) or a principal (p).”

The principal is who will perform the final configuration on the host machines listed in Step 3.
If the name in step 4 is a group name, then the principal who performs the final configuration on the host machines must be a member of the specified group.

```
g (or p)
```

6 Enter your cell-administrator password:

```
cell_admin_password
```

7 At "Do you wish to add NetCrusader/Web support on all of the hosts specified above? (y/n)," For DCE only, answer no:

```
n
```

(This is the same script used by the NetCrusader product, and this option does not apply to DCE. If you anticipate configuring NetCrusaderWeb, you can answer yes, and follow the directions in the *NetCrusaderWeb Installation and Configuration Guide*)

8 Inform the local administrator that the host machines listed in Step 3 above are ready to be configured into the cell.

The local administrators can now configure clients in the cell. See the *PC-DCE Configuration Panel* help file for information about configuring clients. See the *PC-DCE Overview Guide* section 1.3.7 for prerequisites.

2.3.2 Using the Client Configuration Tool

Entegrity provides a command line client configuration tool. If you have multiple clients to configure, this option may be preferable to the graphical PC-DCE configuration tool (the PC-DCE Configuration Panel, accessible from the Windows Control Panel).

The command-line configuration tool provides more flexibility in how you choose to implement it. For example, you can write a program tailored to your environment that calls the command line configuration tool and automatically configures users' systems with the correct client information (see the sample in *Section 2.3.2.1*).

This tool is found in the install_dir/samples directory. There are two files:

- **dcecfgng.exe** — Command-line configuration tool.
- **sample.cf** — Sample configuration file from which **dcecfgng** reads configuration settings. **Sample.cf** describes the values you can input for each option. Use a text editor to customize the settings in the configuration file as your environment requires. To retain the original sample configuration file, save it with another filename and edit the renamed copy.

dcecfgng provides the same client configuration options as the graphical PC-DCE configuration tool (the PC-DCE Configuration Panel, accessible from the Windows Control Panel). If you have questions about any of the command line client configuration options, you can refer to the PC-DCE Configuration Panel online help for information.

To run the command line configuration tool:

1 Customize the sample configuration file. If you are configuring over an existing PC-DCE configuration, make sure that the OnConfigExists setting is set to Proceed.

2 To run the configuration program, from the DOS prompt run **dcecfgng.exe** with the options **-cf** *config file* and **-pw** *cell_admin_password.* For example:

```
c:\ dcecfngu.exe -cf maple -pw -dce-
```

Note that **dcecfgng** requires the *cell_admin_password* for lightweight client configurations as well as full client configurations.

The configuration program may take a few minutes to complete.

2.3.2.1 Sample Use of dcecfgng

One example of using **dcecfgng** to configure multiple client systems is to write a program that runs **dcecfgng** with the command line options. Place the program, along with **dcecfgng** and the **sample.cf** file on a Windows 2000 or Windows NT server that client systems can access.

Clients map to the server and run the program, which executes **dcecfgng** using your customized configuration (**.cf**) file settings. This updates the PC-DCE configuration and Windows registry entries on the client system.

dcecfgng requires the **cell_admin** password to be passed as an argument. If you are concerned about users potentially viewing the password, you can write a program (such as a Visual Basic® program) for users to run, which in turn runs the program that is stored on the server.

CHAPTER 3

Managing Users

This chapter describes basic management of principals, groups, organizations, users, and accounts. It contains the following sections:

3.1 Using the DCE Director

For frequently-performed tasks, such as managing users, groups, and objects, it is easiest to use the DCE Director. The DCE Director is a graphical tool for managing DCE cells. The DCE Director makes it easy to perform management tasks, such as creating, deleting, and modifying user accounts, security groups, and CDS directories. In addition, the DCE Director allows you to access the standard DCE control programs (**rgy_edit**, **cdscp**, **acl_edit**, and **dtscp**), while providing new functions, such as allowing authorized users to preconfigure host machines in a cell and manage user accounts.

The DCE Director includes an enhanced ACL editor, the Visual DCE ACL Editor, which allows you to graphically manage ACLs. You can invoke the Visual DCE ACL Editor directly from the DCE Director or you can use it as a stand-alone tool by clicking its icon in the DCE program group.

For more information on either the DCE Director or the Visual DCE ACL Editor, refer to their respective online help systems.

If you are not using the DCE Director, you can use the DCE command line tools as described in this chapter to perform the same functions.

3.2 Definitions

This section defines many of the terms you will encounter in this chapter. You should understand these terms before attempting to perform the tasks described in this chapter.

Account – An entry in the registry database that defines a principal’s network identity by associating the principal with a group and optional organization, and with related account information such as the password used to authenticate a principal’s identity.

Alias – An alias is an optional alternate name for a primary name. You can assign aliases to principals and groups, but not to organizations. An alias and its associated primary name share the same uid and uuid. You can use an alias on the command line to specify a principal or group.

Because you can create an account for each primary name and each alias, aliases give you the flexibility to establish several accounts for the same principal. For example, suppose that you create a principal with primary name **groucho** and two aliases: **gmarx** and **gm**. You can then create three accounts for the principal **groucho**: one for the primary name and one for each of the name’s aliases. The accounts can use different passwords and can be associated with different access rights, groups and organizations.

For groups, aliases are useful if you want to associate two group names with the same **uid**.

Full Name– You can optionally assign a full name to a principal, group, or organization. A full name typically describes or expands a primary name to allow easy recognition by users. For example, a principal could have a primary name of **jsbach** and a full name of **Johann S. Bach**. A full name is a data field only. You do not use it on the command line to specify a principal, group, or organization.

GID – The UNIX group ID associated with a group.

Group – Named set of principals who can be granted common access rights. Group names are included in access control lists (ACLs) that regulate user access to various server and data objects in the DCE environment.

Object Creation Quota – Attribute associated with a principal that controls the number of registry objects that can be created by the principal. If you allow users to create their own groups, for example, you can use this quota to limit the total number of groups they can create.

Each time a principal creates a registry object, the principal’s object creation quota is decremented by 1. When the object creation quota reaches 0, the principal is prohibited from creating registry objects unless you reset the object creation quota to a number other than 0 by using the **dcecp principal modify** command.

Organization – Named set of users who can be granted common access rights, usually by means of administrative policy. Policies control things like the lifespan of accounts, whether or when account passwords expire, or whether passwords can contain nonalphanumeric characters.

OrgID – The UNIX group ID associated with an organization.

Primary Name – Primary names are assigned to principals, groups, and organizations. A primary name you will typically use when specifying a principal at the command line.

Principal – An entity that can communicate securely with another entity. Principals are represented as entries in the security registry database and include users, servers, and computers. A principal must exist before you can create an account.

Project List – A principal's project list is a list of all the groups in which a principal or alias is a member. When a principal tries to access an object, the principal has the access rights that accrue from membership in every group that is named in the object's ACL. For example, assume the ACL for file X contains two entries: one permits group A write access and one permits group B read access. Then, any principal who is a member of both groups A and B can read and write to file X.

Principals accrue project list access rights only from the groups that are associated with the name or alias with which they log in. For example, assume that a principal named **gustav** is a member of groups A and B. Under the alias **gus**, he is a member of groups C and D. When the principal logs in as **gustav**, the principal accrues access rights from groups A and B only. When the principal logs in with the alias **gus**, the principal accrues access rights from groups C and D only.

User - Refers either to a person who wishes to use DCE services, or the collection of security registry information required for such a person. The registry information required for a user consists of a principal identity and an account.

UUID – Universal Unique ID that identifies an object in the registry database. Normally, you do not have to be aware of UUIDs. They are created and maintained automatically. However, be aware that, although the DCE Security Service prints names and you can access objects by name, it identifies all objects internally by UUID.

UID – UNIX user ID number associated with a user, which the registry uses for compatibility with UNIX programs.

3.3 Name Formats

Names in the registry can contain any characters or digits, except the @ (at sign) and the **:** (colon) character. They must not exceed 1024 characters in length.

3.4 Duplicate Names

You must assign a name to each principal, group, and organization in the registry. Although a principal, a group, and an organization can have the same name, no two principals, groups, or organizations can have the same name. For example, two principals cannot be named **smith**, but a principal can be named **smith**, a group can be named **smith**, and an organization can be named **smith**. You can assign up to three types of names: primary, full, and aliases.

3.5 Managing Users

This section describes how to create and delete users, and show user information.

3.5.1 Creating Users

Use the **dcecp user** command to manage users. This command does the following:

1. Creates a new principal name and adds the principal to a security group and organization. If the security group or organization does not exist when you invoke the operation, you can force its creation by using the **-force** option.
2. Creates an account for the principal and creates the user's password.
3. Adds a directory called **/.:/users/***principalname* to CDS. This directory can store user-specific application location information.
4. Adds an ACL entry to the default ACL which gives the user **rwtci** permissions on the **/.:/users/***principalname* directory. These permissions allow users to insert objects and links, but they cannot delete the directory or administer replication on the directory. Furthermore, users cannot create additional directories unless you give them **w** (write) access to the clearinghouse.

The following example creates a principal name **groucho** and an account with the same name:

```
C:\> dce_login cell_admin -dce-
C:\> dcecp
dcecp> user create groucho -group none -organization none
-fullname {Groucho Marx} -mypwd -dce- -password change.me
```

This example uses group **none** and organization **none** because they exist by default in a new cell.

You can create multiple users by specifying a list of user names as an argument to the **user create** operation. This method poses some limitations, however. All created users will have the same initial password, group name, and organization name. Furthermore, you cannot specify the **fullname** and **uid** attributes since these are unique for each user.

The following example creates several users with a password **change.me**, a group name of **none**, and an organization named **none**:

```
dcecp> user create {groucho harpo chico zeppo} -group none
-organization none -mypwd -dce- -password change.me
```

3.5.2 Showing User Information

The **user show** command returns the principal attributes and ERAs, account attributes, and policies associated with a user. For example:

```
dcecp> user show harpo
```

```
{fullname {Harpo Marx}}
{uid 107}
{uuid 0000006b-80e6-2533-8d00-00802964ff95}
{alias no}
{quota unlimited}
{groups none}
{acctvalid yes}
{client yes}
{created /.../longwood/cell_admin 2000-04-14-11:29:42.000-04:00I-----}
{description {}}
{dupkey no}
{expdate none}
{forwardabletkt yes}
{goodsince 2000-04-14-11:29:42.000-04:00I-----}
{group none}
{home /}
{lastchange /.../longwood/cell_admin 2000-04-14-11:29:42.000-04:00I-----}
{organization none}
{postdatedtkt no}
{proxiabletkt no}
{pwdvalid yes}
{renewabletkt yes}
{server yes}
{shell {}}
{stdtgtauth yes}
nopolicy
```

You can show information about multiple users by specifying a list of user names as an argument to the **user show** command.

3.5.3 Deleting Users

The **dcecp** command **user delete** removes the principal name from the registry and deletes the account and removes the principal from any groups and organizations. The operation also deletes the **/.:/users/***principalname* directory and any contents from CDS.

For example:

```
C:\> dce_login cell_admin -dce-
C:\> dcecp
dcecp> user delete zeppo
```

You can remove multiple users from your cell by specifying a list of user names as an argument to the **user delete** operation, as follows:

```
dcecp> user delete {groucho harpo chico zeppo}
```

If you have permissions in a foreign cell, you can remove one or more users from that cell by specifying the global principal name of the users to be deleted.

For example:

```
dcecp> user delete /.../their_cell.goodco.com/J_Jones
```

3.6 Managing Principals

Normally you use the **dcecp user** command to manage users. However, if you need to manage principals separately (for example, to create a principal without an associated account), this section describes how.

This section describes how to create, delete, and modify principals, and show principal information.

3.6.1 Reserved Principals and Accounts

Some principals and accounts are reserved for use by DCE. You cannot delete or modify reserved principals. You can modify, but not directly delete reserved accounts. A list of reserved principals and accounts follows. In the list *cell_name* is the name of your cell, and *host_principal_name* is the name of the machine principal. The actual form of this name is set during DCE configuration.

- Reserved Principals:
  - **dce-ptgt**
  - **krbtgt/***cell_name*
  - **dce-rgy**
  - *host_principal_name*
- Reserved Accounts:
  - **dce-ptgt none none**
  - **krbtgt/***cell_name* **none none**
  - **dce-rgy none none**
  - *host_principal_name* **none none**

3.6.2 Adding Principals

To add principals to the registry, use the **principal create** command. For example, the following sample command creates a principal with a primary name of **jagger**:

```
dcecp> principal create jagger
```

There are additional attributes you can associate with the principal, including full name and quota.

3.6.2.1 Specifying a Full Name

The **fullname** is a string providing additional information about the principal. Typically, it contains a user's full name. For example, the following command creates a principal and an associated fullname:

```
dcecp> principal create jagger -fullname {Mick Jagger}
```

3.6.2.2 Specifying an Object Creation Quota

The object creation quota is the number of registry objects that this principal can create. Each time a principal creates a registry object, this value is decremented for that principal. To allow a principal to create an unlimited number of registry objects, enter the text string **unlimited**. To prevent a principal from creating any registry objects, enter 0. If you don't enter this argument, the quota defaults to **unlimited**. For example:

```
dcecp> principal create jagger -quota 5
```

NOTE: For an account for a foreign cell (used for intercell communication), the quota is cumulative for all principals who use the account. The quota is *not* per foreign principal. For example, if the quota is 10, the total number of objects that can be created by foreign principals who use the account cannot exceed 10.

3.6.2.3 Creating Multiple Principals

You can create multiple principals with one **principal create** command. To do so, enclose the principal names in braces, separated by spaces. For example, to create the principals **jones**, **watts**, **wyman**, **jagger**, and **richards**, enter the following:

```
dcecp> principal create {jones watts wyman jagger richards}
```

The following sample command creates the principals **jones**, **watts**, **wyman**, **jagger**, and **richards** and assigns each an object creation quota of 100.

```
dcecp> principal create {jones watts wyman jagger richards} -quota 100
```

3.6.3 Showing Principal Information

The **dcecp** command **principal catalog** displays a list of the principals in the cell. For example:

```
dcecp> principal catalog

/.../longwood/nobody
/.../longwood/root
/.../longwood/daemon
/.../longwood/sys
/.../longwood/bin
/.../longwood/uucp
/.../longwood/who
/.../longwood/mail
/.../longwood/tcb
/.../longwood/dce-ptgt
/.../longwood/dce-rgy
/.../longwood/cell_admin
/.../longwood/krbtgt/longwood
/.../longwood/hosts/darwin.entegrity.com/self
/.../longwood/hosts/darwin.entegrity.com/cds-server
/.../longwood/hosts/darwin.entegrity.com/gda
/.../longwood/hosts/chest.entegrity.com/self
```

```
/.../longwood/groucho
/.../longwood/harpo
/.../longwood/jagger
/.../longwood/hm
```

The **dcecp** command **principal show** *principal* displays attribute information about the principal. For example:

```
dcecp> principal show harpo
```

```
{fullname {Harpo Marx}}
{uid 107}
{uuid 0000006b-80e6-2533-8d00-00802964ff95}
{alias no}
{quota unlimited}
{groups none}
```

3.6.4 Modifying Principals

You can change a principal's primary name and other information related to the principal. The change is reflected in the membership lists of all the groups and organizations in which the principal is a member.

Additionally, you can change a primary name to an alias and an alias to a primary name. If you change a primary name to an alias and do not make an alias the primary name, operations that return names choose one of the aliases at random.

3.6.4.1 Changing the Primary Name

Use the **dcecp principal rename** command to change a primary name. Enter the command in the following form:

principal rename *old_name* **-to** *new_name*

old_name – primary name of the principal to be changed.

new_name – new primary name of the principal.

The following example shows the **principal rename** command used to change a full name from **smit** to **smith**:

```
dcecp> principal rename smit -to smith
```

3.6.4.2 Changing Principal Information

Use the **dcecp principal modify** command to change any principal information except the **uid** and **uuid**. The following example shows the **principal modify** command used to change principal **jones**'s object creation quota to 10.

```
dcecp> principal modify jones -quota 10
```

3.6.4.3 Adding an Alias to a Principal

Use the **dcecp** command **principal create** to create an alias and associate it with a principal.

First use the **principal show** command to obtain the **uid** of the principal to which you are adding an alias:

```
dcecp> principal show harpo
{fullname {Harpo Marx}}
{uid 107}
{uuid 0000006b-80e6-2533-8d00-00802964ff95}
{alias no}
{quota unlimited}
{groups none}
```

Then issue the principal create command specifying the principal's **uid** and the **-alias yes** option:

```
dcecp> principal create hm -uid 107 -alias yes
```

3.6.5 Deleting Principals and Aliases

If you delete a principal or an alias, the system automatically deletes any accounts for that principal or alias. Be aware that deleting a principal or a principal's alias could orphan the objects that are owned by the principal (Refer to *Section 3.6.6*).

The following example shows the **principal delete** command used to delete the principal named **mahler**:

```
dcecp> principal delete mahler
```

You can delete multiple principals or aliases with one **principal delete** command. To do so, enclose the principal names in braces, separated by spaces. For example, to delete the principals **bach**, **britten**, and **richards**, you would enter the following:

```
dcecp> principal delete {bach britten richards}
```

3.6.6 Recovering Orphaned Objects

If you delete a principal from the registry, you also delete the principal's UUID. Any objects (files, programs) that are owned by the principal are associated with an orphaned UUID; that is, a UUID with no corresponding name. This means that the object is now owned by a deleted principal. If no other principals were previously given access to the object, the object cannot be accessed.

To solve this problem, use the **dcecp principal create** command with the **-uuid** option to associate the UUID with a name and thus adopt the orphaned object. UUIDs are assigned automatically when the object is created by using the DCE control program's **principal create** command. Therefore, you cannot simply add a new user and acquire a previously used UUID. You must execute the **dcecp principal create** command with the **-uuid** option for this purpose.

For example, an **acl show** of the object *grade* shows that only the user `fred` has privileges to the object:

```
dcecp> acl show -e /.:/grade
{user fred rwdtc}
```

If the principal fred is deleted, the object is an orphan:

```
dcecp> principal delete fred
dcecp> acl show -e /.:/grade
{user 00000080-d459-25b4-8000-0000c0987001 rwdtc}
```

Use the UUID now displayed in place of the name to create a new principal for the orphaned object. You can use the same name or a different name:

```
dcecp> principal create wilma -uuid 00000080-d459-25b4-8000-0000c0987001
dcecp> acl show -e /.:/grade
{user wilma rwdtc}
```

3.7 Managing Groups and Organizations

Groups provide a convenient way to control access rights for a set of users with the same security requirements. When you edit an object ACL to configure permissions, specifying the group name grants those permissions to all group members. (The exception is if the object ACL contains an entry for a specific user; in this case, the user permissions override the permissions for any group of which this user is a member.)

Use organizations to simplify policy management (policy regulates things like account and password lifetimes and password format). An organization’s policies override the registry default policies if the organization’s policies are more restrictive.

This section describes how to create and delete groups and organizations, and show group and organization information.

3.7.1 Adding Groups and Organizations

Use the **dcecp group create** command to add groups and the **dcecp organization create** command to add organizations. When you add a group or organization, you must specify the group’s or organization’s primary name.

Note that, when you use the **dcecp group create** command and **dcecp organization create** command, you can create multiple groups or organizations with one command in the same way that you can create multiple principals.

3.7.1.1 Adding a Group

The following example shows how to add a group named **symphonists** to the **registry**:

```
dcecp> group create symphonists
```

3.7.1.2 Adding an Organization

The following example shows how to add an organization named **classic** to the registry:

```
dcecp> organization create classic
```

3.7.2 Showing Group and Organization Information

The **dcecp** commands **group catalog** and **organization catalog** display lists of the groups and organizations in the cell. For example:

```
dcecp> group catalog
```

```
/.../longwood/nogroup
/.../longwood/system
/.../longwood/daemon
/.../longwood/uucp
/.../longwood/bin
/.../longwood/kmem
/.../longwood/mail
/.../longwood/tty
/.../longwood/none
/.../longwood/tcb
/.../longwood/acct-admin
/.../longwood/subsys/dce/sec-admin
/.../longwood/subsys/dce/cds-admin
/.../longwood/subsys/dce/dts-admin
/.../longwood/subsys/dce/dskl-admin
/.../longwood/subsys/dce/cds-server
/.../longwood/subsys/dce/dts-servers
/.../longwood/subsys/dce/audit-admin
/.../longwood/subsys/dce/dced-admin
```

The **dcecp** commands **group show** *group* and **organization show** *organization* display attribute information about groups and organizations. For example:

```
dcecp> group show none
```

```
{alias no}
{gid 12}
{uuid 0000000c-7bda-2533-9f01-00802964ff95}
{inprojlist yes}
{fullname {}}
```

3.7.3 Modifying Groups and Organizations

For groups and organizations, you can change the primary name and full name. In addition, for groups you can change whether or not the group can appear in project lists, and for organizations you can change policy.

Use the **dcecp group modify** command to change groups. The following example shows the use of this command with the **-inprojlist** option to change the group **symphonist**’s project list inclusion property from **yes** (include on project lists) to **no** (prohibit from project lists).

```
dcecp> group modify symphonists -inprojlist no
```

Use the **dcecp group rename** command to change a group’s primary name or the **dcecp organization rename** command to change an organization’s primary name. These commands have the following form:

group rename *old_name* **-to** *new_name*

organization rename *old_name* **-to** *new_name*

where:

old_name – Primary name of the group or organization to be changed.

new_name – New primary name of the group or organization.

The following example shows the **group rename** command used to change a full name from **symphonists** to **symphonists7**:

```
dcecp> group rename symphonists -to symphonists7
```

Note that, if you change a primary name, that change is reflected in the membership lists of all the groups and organizations in which the group or organization is listed as a member.

3.7.4 Deleting Groups and Organizations

If you delete a group or organization, you also automatically delete any accounts that use the group or organization. For example, if you delete the group **symphonists**, you also automatically delete the accounts **vivaldi symphonists baroque** and **mozart symphonists classic**.

NOTE: The default groups **none** and **nogroup** and the default organization **none** represent users that have either not yet been assigned to a group or organization or have been chosen not to be assigned to any group or organization. DCE needs these groups — do not delete them.

Use the **dcecp group delete** to delete groups and the **dcecp organization delete** command to delete organizations. The following example shows the **group delete** command being used to delete the group **symphonists**:

```
dcecp> group delete symphonists
```

The next example shows the **organization delete** command being used to delete the organization **classic**:

```
dcecp> organization delete classic
```

Note that you can delete multiple groups or organizations with a single **group delete** or **organization delete** command by including the names to delete in braces and separated by spaces just as you would to delete multiple principals.

3.7.5 Maintaining Membership Lists

Each group and organization has a membership list, which lists the principals that are members of the group or organization. Use the **dcecp group add** command to add members to the membership list and the **dcecp group remove** command to remove members from the list.

If you delete a member from a group or organization, any accounts for the deleted member that are associated with the group or organization are also deleted. For example, if you delete the principal **mahler** from the group **symphonists**, the account **mahler symphonists classic** is also deleted.

Note that the deleting of a principal from a group or organization can affect the principal’s rights to objects. This change takes effect only when the principal’s ticket-granting ticket is renewed.

3.7.5.1 Effects of Account Creation on Membership Lists

When you create accounts, the principal for whom the account is created must be a member of the group or organization that is named in the account. For example, if you create the account **mahler symphonists classic**, the principal **mahler** must be a member of the **symphonists** group and the **classic** organization.

The **dcecp** command recognizes this requirement and, if you have the permissions to add to the group or organization, tries to add the principal to the group and organization. For example, assume that the principal **mahler** is not a member of either the group **symphonists** or the organization **classic**. If you have the proper permissions when you create the account **mahler symphonists classic**, the **account create** command automatically adds **mahler** to the **symphonists** and **classic** membership lists so that you can create the account in one step.

However, if you do not have the required permissions, the command fails and displays a message like the following:

```
Not authorized to perform operation
```

3.7.5.2 Adding and Deleting Group Members

The following example shows the use of the **dcecp group add** command with the **-member** option to add **mahler** to the group **symphonists** and delete **strauss** from the group **symphonists**:

```
dcecp> group add symphonists -member mahler
dcecp> group remove symphonists -member strauss
```

You can add and remove multiple members with one **group add** or **group remove** command. To do so, enclose the member names in quotes, separated by spaces. For example, to add the principals **bach**, **britten**, and **mccartney** to the group **symphonists**, you would enter the following:

```
dcecp> group add symphonists -member {bach britten mccartney}
```

3.7.5.3 Displaying Membership Lists

To display the members of a group or organization, use the **dcecp** command **group list** *group* or **organization list** *organization*. For example:

```
dcecp> group list none

/.../longwood/dce-ptgt
/.../longwood/dce-rgy
/.../longwood/krbtgt/longwood
/.../longwood/cell_admin
/.../longwood/hosts/darwin.entegrity.com/self
/.../longwood/hosts/darwin.entegrity.com/gda
/.../longwood/hosts/darwin.entegrity.com/cds-server
/.../longwood/hosts/chest.entegrity.com/self
/.../longwood/groucho
/.../longwood/harpo
```

3.8 Managing Accounts

Registry accounts define a network identity by associating a principal with a group, an organization, and related account information, such as the password that is used to authenticate a principal's identity. You must create a registry account for any principal that engages in communications across the network, regardless of whether the communications are authenticated. The following types of principals require registry accounts:

- Each human user who accesses objects across the network. You can use the DCE Director to create accounts for users, or you can use the **dcecp user** command (*Section 3.5 on page 30*). If the principal, group, and organization already exist, you can use the **dcecp account** command as described in this section.

- Each server that accesses objects across the network and runs under its own identity, not the identity of the principal who started it. The PC-DCE server configuration program automatically creates accounts for PC-DCE servers.

- Each machine in the network. The PC-DCE configuration program creates the required account for each machine running PC-DCE software.
- Any cell with which you engage in authenticated cross-cell communications. (Refer to *Chapter 7* for instructions on creating an account for a foreign cell.)

3.8.1 User Accounts

User accounts are associated with the user's password and information that is used when the user logs into DCE. Account information includes such things as the principal's home directory and login shell, and authentication policy, which defines parameters that help control a principal's access to DCE.

3.8.2 Creating an Account

Normally, to create a user account you use the DCE Director or the **dcecp user create** command, both of which create both a principal and associated account. If you want to create an account for an existing human user principal, use the **dcecp account create** command. For example:

1 Associate the principal with a group and organization:

```
C:\> dce_login cell_admin -dce-
C:\> dcecp
dcecp> group add straight_men -member zeppo
dcecp> organization add entertainers -member zeppo
```

2 Create the account:

```
dcecp> account create zeppo -group straight_men
-organization entertainers -mypwd -dce- -password change.me
```

When creating an account, you can specify any of the account attribute values that you do not wish to default. The following section (*Section 3.8.2.1*) describes these attribute values.

3.8.2.1 Account Attribute Values

When creating an account, you can specify any of the account attribute values that you do not wish to default. Refer to *Table 3-1*.

Table 3-1: Attribute Options to Create Accounts

Option	Meaning
-acctvalid {yes|no}	A flag that determines account validity. If you set this flag to **no**, the account is invalid and the account principal cannot log into the account. The default is **yes**.
-client {yes|no}	A flag that indicates whether or not the account is for a principal that can act as a client. If you set this flag to **yes**, the principal is able to log into the account and acquire tickets for authentication. The default is **yes**.

Table 3-1: Attribute Options to Create Accounts (Continued)

Option	Meaning
-description *string*	A text string in Portable Character Set (PCS) format that is typically used to describe the use of the account. No default.
-dupkey {yes|no}	A flag that determines if tickets issued to the account's principal can have duplicate keys. The default is **no**.
-expdate	The date (in ISO timestamp format *YYYY-MM-DD-hh:mm:ss*) on which the account expires. To renew an account after it expires, change the date. The default is **none**, meaning the account never expires.
-forwardabletkt {yes|no}	A flag determining whether a new ticket-granting ticket with a network address that differs from the present TGT's network address can be issued to the account's principal. (The **-proxiabletkt** attribute performs the same function for service tickets.) The default is **yes**.
-goodsince *date*	The date and time (in ISO timestamp format *YYYY-MM-DD-hh:mm:ss*) that the account was last known to be in an uncompromised state. Any tickets granted before this date are invalid. When the account is created, the **-goodsince** attribute is set to the current date. Control over this date is especially useful if you know that an account's password was compromised. Changing the password can prevent the unauthorized principal from accessing the system again by using that password, but does not prevent the principal from accessing the system components for which tickets were obtained fraudulently before the password was changed. To eliminate the principal's access to the system, the tickets must be canceled. Set the **-goodsince** attribute to the date and time the compromised password was changed to invalidate all tickets issued before that time and eliminate the unauthorized principal's system access.
-group *group_name*	The name of the group that is associated with the account. This attribute must be supplied to create an account; there is no default.
-home *dir_name*	The directory in which the principal is placed at login. No default.
-organization *org_name*	The name of the organization that is associated with the account. This attribute must be supplied to create an account; there is no default.
-password *password*	The required password for the account in plaintext. The system encrypts the password you supply. No default.
-postdatedtkt {yes|no}	A flag that determines whether or not tickets with a start time in the future can be issued to the account's principal. The default is **no**.
-proxiabletkt {yes|no}	A flag determines whether or not a new ticket with a different network address than the present ticket can be issued to the account's principal. (The **-forwardabletkt** attribute option performs the same function for ticket-granting tickets.) The default is **no**.

Table 3-1: Attribute Options to Create Accounts (Continued)

Option	Meaning
-pwdvalid {yes|no}	A flag that determines whether the current password is valid. If this flag is set to **no**, the account password has expired and the principal will be prompted to change it the next time that the principal logs into the account. The default is **yes**.
-renewabletkt {yes|no}	The Kerberos V5 renewable ticket feature is not currently used by DCE; any use of the renewable ticket attribute is unsupported at the present time.
-server {yes|no}	A flag that indicates whether or not the account is for a principal that can act as a server. If the account is for a server that engages in authenticated communications, set this flag to **yes**. The default is **yes**.
-shell *path_to_shell*	The shell that is executed when a principal logs in.
-stdtgtauth {yes|no}	A flag that determines whether or not tickets issued to the account’s principal can use the ticket-granting-ticket authentication mechanism. The default is **yes**.
-maxtktlife *hours*	The maximum ticket lifetime. This is the maximum amount of time in hours that a ticket can be valid. When a client requests a ticket to a server, the lifetime granted to the ticket takes into account the **maxtktlife** attribute value for both the server and the client. In other words, the lifetime cannot exceed the shorter of the server’s or client’s maximum ticket lifetime. If you do not specify a **maxtktlifetime** attribute value for an account, the **maxtktlifetime** attribute value defined for the registry authorization policy is used.
-maxtktrenew *hours*	The maximum ticket renewable. This is the amount of time in hours before a principal’s ticket-granting ticket expires and that principal must log into the system again to reauthenticate and obtain another ticket-granting ticket. The lifetime of the principal’s service tickets can never exceed the lifetime of the principal’s ticket-granting ticket. The shorter you make Maximum Certificate Renewable, the greater the security of the system. However, since principals must log in again to renew their ticket-granting ticket, the time needs to take into consideration user convenience and the level of security required. If you do not specify a **maxtktrenew** attribute value for an account, the **maxtktrenew** attribute value defined for the registry authorization policy is used.

3.8.3 Showing an Account

To view a list of accounts, use the **dcecp account catalog** command. For example:

```
dcecp> account catalog
```

```
/.../longwood/nobody
/.../longwood/root
/.../longwood/daemon
/.../longwood/uucp
/.../longwood/bin
/.../longwood/dce-ptgt
/.../longwood/dce-rgy
/.../longwood/krbtgt/longwood
/.../longwood/cell_admin
/.../longwood/hosts/darwin.entegrity.com/self
/.../longwood/hosts/darwin.entegrity.com/cds-server
/.../longwood/hosts/darwin.entegrity.com/gda
/.../longwood/hosts/banks.entegrity.com/self
/.../longwood/groucho
/.../longwood/harpo
/.../longwood/zeppo
```

To view an account’s attributes, use the **dcecp account show** command. For example:

```
dcecp> account show zeppo
```

```
{acctvalid yes}
{client yes}
{created /.../longwood/cell_admin 2000-04-16-11:05:15.000-04:00I-----}
{description {}}
{dupkey no}
{expdate none}
{forwardabletkt yes}
{goodsince 2000-04-16-11:05:15.000-04:00I-----}
{group none}
{home /}
{lastchange /.../longwood/cell_admin 2000-04-16-11:05:15.000-04:00I-----}
{organization none}
{postdatedtkt no}
{proxiabletkt no}
{pwdvalid yes}
{renewabletkt yes}
{server yes}
{shell {}}
{stdtgtauth yes}
```

3.8.4 Modifying an Account

To modify an attribute, use the **dcecp account modify** command. The following example changes the expiration date on account **zeppo**:

```
dcecp> account modify zeppo -expdate 1999-12-10-00:00:00 -mypwd -dce-
```

3.8.5 Deleting an Account

To delete an account, use the **dcecp account delete** command. The following example deletes account **zeppo**:

```
dcecp> account delete zeppo
```

If you delete a group or organization, you will also automatically delete any accounts that specify this group or organization as the primary group or organization.

You can delete multiple accounts with one **account delete** command. To do so enclose the names of the account principals in braces, separated by spaces. For example, to delete accounts for **bach**, **britten**, and **mahler**, you would enter:

```
dcecp> account delete {bach britten mahler}
```

CHAPTER 4

Authentication

This chapter describes authentication topics and related tasks. It contains the following sections:

For a discussion of Entegrity's Co-Authentication Service (CAS) please refer to the *Guide to CAS*.

4.1 Understanding DCE Authentication

This section provides some general information on DCE authentication useful to those who need to administer DCE cells.

4.1.1 DCE Authentication Process

The following is a very general description of how DCE authentication works.

When you create or modify an account for a principal, you supply a password for that principal. The security service uses the password to derive an authentication key, which it stores in the registry.

When a principal logs into DCE, the security client on the principal's system uses the password supplied by the principal to derive the principal's authentication key. This key is used by the security service to authenticate the principal (that is, to guarantee the principal's identity) as follows:

1. The security client does the following:

   - Uses the password to derive the principal's authentication key
   - Prepares a login request, and encrypts part of it using the authentication key
   - Forwards the request to the security service

2. The security service does the following:
    - Obtains the registry's copy of the principal's authentication key
    - Attempts to decrypt the login request with this key

If the decryption succeeds, the keys are the same, which means the password supplied by the principal is valid. Authentication succeeds and login is allowed.

If the decryption fails, the keys must be different, which means the password supplied by the principal is invalid. Authentication fails and login is denied.

4.1.2 OSF DCE Version 1.1 Authentication Enhancements

OSF DCE Version 1.1 authentication addresses certain security deficiencies in the Kerberos V5 authentication protocols, which are used as the basis for the DCE authentication protocol in versions previous to OSF DCE Version 1.1. These deficiencies result from:

- The security server responding to client login requests without verifying that the user knows the password
- The use of user passwords, which are notoriously weak, to encrypt plaintext data that is then sent across the network

These practices are subject to attacks in which the attacker obtains network transmissions and proceeds to attack them offline to elicit users' passwords.

OSF DCE Version 1.1 reduces the likelihood of such attacks succeeding by providing for:

- Preauthentication of principals making login requests (that is, by having the security service verify the identity of the requestor before responding to the request)
- The use of strong keys to encrypt all network transmissions involving validation between security clients and servers

4.1.2.1 Preauthentication Protocols

Three preauthentication protocols provide three levels of decreasingly strict preauthentication. You can control the minimum level of preauthentication that the security server will accept by attaching an instance of the *pre_auth_req* ERA (Extended Registry Attribute) to the principal, as described in the following section.

The preauthentication protocols are as follows:

- **Third-party protocol** – Provides the highest level of security. This is the highest level of DCE preauthentication and provides the most protection against the attacks previously described. With third-party preauthentication, all authentication data sent over the network is encrypted with a “strong” random key known only to the local machine principal and the DCE Security Service.

  OSF DCE Version 1.1 clients always construct authentication requests with this protocol, except in cases where they are unable to do so because the machine session key, which is required to construct third-party requests, is unavailable (for example, at cell startup).

- **Timestamps protocol** – Provides an intermediate level of security. The timestamps protocol protects against attackers masquerading as a security client and attacking replies from the DCE Authentication Service, but is still vulnerable to attacks by processes capable of monitoring the network.

  Specify timestamps preauthentication only for principals (such as cell administrators and noninteractive principals) who must be able to operate when the client is unable to construct a third-party authentication request as previously described. In these cases, the client constructs and forwards a timestamps login request.

- **OSF DCE Version 1.0 (Kerberos V5) protocol** – Authenticates pre-OSF DCE Version 1.1 clients only, and provides no preauthentication security.

  Specify the OSF DCE Version 1.0 protocol only to enable OSF DCE Version 1.1 servers to accept login requests from pre-OSF DCE Version 1.1 clients.

4.1.2.2 Preauthentication Interoperability Between DCE Versions

Table 4-1 describes how login requests are handled when OSF DCE Version 1.1 clients and servers interoperate with pre-OSF DCE Version 1.1 clients and servers in a single cell.

Table 4-1: OSF DCE Version 1.1/Pre-OSF DCE Version 1.1 Authentication Interoperation

Login Request Type	Pre-1.1 Server Response	1.1 Server Response
OSF LDCE Version 1.0 From pre-OSF DCE Version 1.1 clients only.	Returns OSF DCE Version 1.0 (unpreauthenticated) response.	If no ERA exists, or existing ERA has value=0 (NONE), returns OSF DCE Version 1.0 (unpreauthenticated) response. Otherwise, rejects login request.
TIMESTAMPS From OSF DCE Version 1.1 clients only.	Server ignores preauthentication data in request and returns pre-OSF DCE Version 1.1 (unpreauthenticated) response.	If no ERA exists, or existing ERA has value=0 (NONE) or value= 1 (TIMESTAMPS), returns OSF DCE Version 1.1 TIMESTAMPS response. If existing ERA has value=2 (THIRD-PARTY), rejects login request.
THIRD-PARTY From OSF DCE Version 1.1 clients only.	Server ignores preauthentication data in request and returns pre-OSF DCE Version 1.1 (unpreauthenticated) response.	Returns OSF DCE Version 1.1 THIRD-PARTY response.

4.1.3 Privilege Attributes

After a principal is authenticated, the security service obtains the principal's privilege attributes. Privilege attributes include the principal's network identity, the groups in which the principal is a member, and any extended attributes associated with the principal. Privilege attributes determine the principal's rights to any objects that principal attempts to access.

4.1.4 Ticket-Granting Tickets and Service Tickets

A *ticket-granting ticket* (TGT) allows a principal to request and receive tickets to DCE services. The tickets that let principals access DCE services are called *service tickets*.

Both ticket-granting tickets and service tickets have lifetimes that are determined by the settings for individual accounts and registry policies and properties. When a principal's ticket-granting ticket expires, the principal is no longer considered an authenticated user. To remedy this, the principal must reauthenticate by running the **kinit** command or by logging in again to DCE.

If you flag an account as able to renew service tickets, the principal's service tickets are renewed automatically by the authentication service, requiring no action on the principal's part. Note, however, that the lifetime allocated to a service ticket can never exceed the time remaining on the principal's ticket-granting ticket.

4.2 Managing DCE Authentication

This section describes how to manage a principal's pre-authentication protocol settings and tickets.

4.2.1 Managing Preauthentication Protocols

You manage preauthentication for a given principal by attaching an instance of the *pre_auth_req* ERA to the principal and specifying a value to indicate *the lowest level protocol* the DCE Security Service should accept for the principal, as follows:

- **0 (NONE)** – Specifies that the security service should accept, from this principal, login requests using any of the three protocols (including the OSF DCE Version 1.0 protocol.)

  NOTE: Failing to attach an instance of the *pre_auth_req* ERA to a principal is equivalent to specifying **0 (NONE)**.

- **1 (PADATA-ENC-TIMESTAMPS)** – Specifies that the DCE Security Service should accept, from this principal, login requests using either the timestamp or third-party protocol.

- **2 (PADATA-ENC-THIRD-PARTY)** – Specifies that the only login requests the DCE Security Service will accept from this principal are those using the third-party protocol.

When the authentication service receives a login request for a principal, it always attempts to respond using the same protocol as the request, unless the *pre_auth_req* ERA value for that principal forbids it to do so.

4.2.1.1 Creating a Principal with the pre_auth_req ERA

Use the **principal create** *principal* **-attribute {***attribute_list***}** command to create a principal and attach the **pre_auth_req** ERA. For example:

```
dcecp> principal create fillmore -attribute {pre_auth_req 2}
```

4.2.1.2 Viewing a Principal's ERAs

Use the **principal show** *principal* **-xattrs** command to display a principal's ERAs. For example:

```
dcecp> principal show fillmore -xattrs
{pre_auth_req 2}
```

4.2.1.3 Adding the pre_auth_req ERA to an Existing Principal

Use the **principal modify** *principal* **-add {***attribute_list***}** command to attach the **pre_auth_req** ERA to an existing principal. For example:

```
dcecp> principal modify cell_admin -add {pre_auth_req 2}
```

4.2.1.4 Modifying a Principal's pre_auth_req ERA Value

Use the **principal modify** *principal* **-change {***attribute_list***}** command to change the **pre_auth_req** value. For example:

```
dcecp> principal modify fillmore -change {pre_auth_req 2}
```

4.2.2 Managing Privilege Attributes and Tickets

This section describes how to manage a principal's pre-authentication protocol settings and tickets.

4.2.2.1 Viewing Privilege Attributes and Tickets

Use the dcecp **klist** command to display a principal's current tickets and privilege attributes. The **klist** command displays three types of information:

- Privilege attributes
- Expiration information
- service ticket information

The First Part of the klist Display—Privilege Attributes

The **klist** command displays a principal's privilege attributes. This display lists the fully qualified principal name followed by the UUIDs and names of the cell, the principal name, and all the groups of which the principals is a member. For example:

```
dcecp> klist
DCE Identity Information:
        Global Principal: /.../longwood/fred
        Cell:      004127b0-916c-14bf-a883-00802964ff95 /.../longwood
        Principal: 00000064-916c-24bf-a800-00802964ff95 music/fred
        Group:     0000000c-916d-24bf-a801-00802964ff95 composers
```

The Second Part of the klist Display—Expiration Dates and Times

The second part of the **klist** display shows the dates and time that the principal's ticket-granting ticket, account, and password expire:

- The first line shows the date and time the ticket-granting ticket expires.
- The second line shows when the principal's account expires. If the account expires, the principal will be unable to log into DCE.
- The third line shows the date the principal's password expires. If the password expires, the principal will be unable to log into DCE.

For example:

```
Identity Info Expires: 2000/02/13:00:56:13
Account Expires:       2000/12/31:12:00:00
Passwd Expires:        2000/06/30:12:00:00
```

The Third Part of the klist Display—Tickets

The third part of the **klist** display shows the principal's ticket information and the name of the principal's ticket cache.

The first three tickets labeled **Server** are the tickets used after the principal logged in and obtained privilege attributes. The remaining tickets labeled **Client** show the principal's ticket-granting ticket and service tickets.

In the listing for each ticket after the word **Client**, the display shows the name of the privilege server, a server that grants privilege attributes after the principal's identity has been authenticated by the security service. The name of the server to which the principal has tickets is shown after the **Server** entry, and the dates and times these tickets are valid are shown on the following line.

For example, in the following sample display, the last line shows that the principal has a ticket to the server named **dce-rgy**. The lifetime of this ticket is from 2:56 and 15 seconds p.m. on 2/12/98 to 4:56 and 15 seconds p.m. on 2/12/98. (The time is shown in 24-hour format.)

```
Kerberos Ticket Information:
Ticket cache: C:/PCDCE32/opt/dcelocal/var/security/creds/dcecred_00f58600
Default principal: fred@longwood
Server: krbtgt/longwood@longwood
        valid 2000/02/12:14:56:13 to 2000/02/13:00:56:13
Server: dce-rgy@longwood
        valid 2000/02/12:14:56:13 to 2000/02/13:00:56:13
```

```
Server: dce-ptgt@longwood
        valid 2000/02/12:14:56:15 to 2000/02/12:16:56:15
Client: dce-ptgt@longwood       Server: krbtgt/longwood@longwood
        valid 2000/02/12:14:56:15 to 2000/02/12:16:56:15
Client: dce-ptgt@longwood       Server: dce-rgy@longwood
        valid 2000/02/12:14:56:15 to 2000/02/12:16:56:15
```

4.2.2.2 Destroying Tickets

Use the **kdestroy** command to invalidate the tickets that a principal has acquired.

When the principal logs out, the principal's tickets are not destroyed; they remain valid until they expire. Users may want to use **kdestroy** just before they log out to ensure that no valid tickets remain.

4.3 Managing Invalid Login Handling

When you specify a preauthentication level of 2 (THIRD-PARTY) for a principal, the security server is able to detect and track invalid login attempts for that principal. This makes it possible for administrators to limit the possible impact of password guessing attacks by:

- Setting a limit to the number of successive invalid login attempts before the principal's account is disabled. (A successful login resets the counter.)
- Specifying the period of time the principal's account will be disabled once that limit is reached.

You do this by attaching instances of two ERAs (*max_invalid_attempts* and *disable_time_interval*) to the principal:

- **max_invalid_attempts** – Number of successive invalid login attempts allowed before the security server disables the account.
- **disable_time_interval** – Number of seconds the principal's account should be disabled once **max_invalid_attempts** is exceeded.

If the user's account is disabled due to exceeding the limit set by **max_invalid_attempts**, the login response the user sees is identical to the response for an invalid password.

4.3.1 Creating a Principal with Attached Invalid Login ERAs

Use the **principal create** *principal* **-attribute {***attribute_list***}** command to create a principal and attach the invalid login ERAs. For example:

```
dcecp> principal create fillmore -attribute {{pre_auth_req 2}
{max_invalid_attempts 3} {disable_time_interval 60}}
```

4.3.2 Viewing a Principal's ERAs

Use the **principal show** *principal* **-xattrs** command to display a principal's ERAs. For example:

```
dcecp> principal show fillmore -xattrs
{pre_auth_req 2}
{max_invalid_attempts 3}
{disable_time_interval 60}
```

4.3.3 Adding Invalid Login ERAs to an Existing Principal

Use the **principal modify** *principal* **-add {***attribute_list***}** command to attach the **pre_auth_req** ERA to an existing principal. For example:

```
dcecp> principal modify taylor -add {{pre_auth_req 2}
{max_invalid_attempts 3} {disable_time_interval 60}}
```

4.3.4 Modifying a Principal's Invalid Login ERA Values

Use the **principal modify** *principal* **-change {***attribute_list***}** command to change the **pre_auth_req** value. For example:

```
dcecp> principal modify cell_admin -change {disable_time_interval 300}
```

4.3.5 OSF DCE Version 1.1 Limitation

In OSF DCE Version 1.1, the invalid login handling functionality accurately tracks login activity in a cell with one master and no replicas, but does not keep accurate counts in replicated cells. This is because:

- Login attempts in a replicated cell are randomly assigned to either a master or replica.
- There is at present no mechanism for replicas to communicate to the master and, therefore, no way for the master to maintain an accurate count.

4.4 Managing Password Format Policies

You can restrict the formats used for newly created or modified passwords by setting the following policy attributes:

- **pwdminlen** *integer* – Minimum password length. A value of 0 specifies no minimum length. The default is 0.
- **pwdspaces** {**yes** | **no**} – Whether or not the password can be all spaces. The default is yes.
- **pwdalpha** {**yes** | **no**} – Whether or not the password can contain only alphanumeric characters. The default is yes.

For a given account, password format policy consists of the combination of the strongest settings from the registry and from the account's primary organization.

The security service enforces the policy when you create or modify a password. Therefore, if you change the policy, existing passwords are not subjected to the change.

You can also use the password strength server to enforce formats for selected accounts. For information on the password strength server, refer to *Section 4.5*.

4.4.1 Viewing Password Format Settings

To view policy settings, including password formats, for the entire registry, use the **registry show** *registry* **-policies** command. For example:

```
dcecp> registry show /.:/subsys/dce/sec/master -policies
{acctlife unlimited}
{maxtktlife +1-00:00:00.000I-----}
{maxtktrenew +28-00:00:00.000I-----}
{pwdalpha yes}
{pwdexpdate none}
{pwdlife unlimited}
{pwdminlen 6}
{pwdspaces yes}
```

To view policy settings for an organization, use the **organization show** *organization* **-policies** command. For example:

```
dcecp> org show zephyr -policies
{acctlife unlimited}
{pwdalpha no}
{pwdexpdate none}
{pwdlife unlimited}
{pwdminlen 0}
{pwdspaces yes}
```

4.4.2 Changing Password Format Settings

To change a policy attribute in the registry, use the dcecp **registry modify** command. For example, the following command changes the minimum password length to six for the entire registry:

```
dcecp> registry modify /.:/subsys/dce/sec/master -change {pwdminlen 6}
```

To change a policy attribute in an organization, use the dcecp **organization modify** command. For example, the following command forbids the use of passwords consisting of all spaces for organization zephyr:

```
dcecp> org modify zephyr -change {pwdspaces no}
```

4.5 Using the Password Strength Server

The password strength server performs the following functions for specified principals:

- Enforcement of password format policy – The password strength server enforces either the existing password policy, as defined by registry and organization attributes (*Section 4.4 on page 54*), or a separate policy defined by arguments passed to the password strength server at startup time.
- Password generation – The password strength server generates randomized passwords.

You specify which principals are subject to the password strength server by attaching ERAs to the principals specifying the service required of the password strength server and a binding to the server.

4.5.1 Configuring the Password Strength Server

The password strength server is an additional DCE server provided by PC-DCE. You configure the password strength server using the PC-DCE Service Panel. For instructions on configuring a password strength server, refer to the online help associated with the PC-DCE Configuration Panel.

NOTE: To protect password security, and to optimize performance, the password strength server should run on the same machine as the master DCE security server.

Once you configure the password strength server, it runs automatically as part of the PC-DCE service. You can control it using the PC-DCE Service Panel along with the rest of the DCE processes. You can also run it from the command line as described in *Section 4.5.3 on page 59*.

If you want to modify the password strength server default behavior, the password strength server source code is available from The Open Group. For information, visit **http://www.opengroup.org/tech/dce/mall/free_dce.htm.**

4.5.2 Configuring Principals to Use the Password Strength Server

To configure a principal to use the password strength server, attach instances of the *pwd_val_type* and *pwd_mgmt_binding* ERAs:

- **pwd_val_type** *integer* – Specifies password creation options for the principal as follows:

  0 (NONE) – Specifies that the principal's password is subject only to checking by the standard security service. Specifying **0** is equivalent to not attaching an ERA instance to the principal.

  1 (USER_SELECT) – Specifies that the principal must supply a password. The password strength server strength-checks the password (enforces password policy).

  2 (USER_CAN_SELECT) – Specifies that the principal can either supply a password or request a randomized password from the password strength server. The password strength server strength-checks the password (enforces password policy).

  3 (GENERATION_REQUIRED) – Specifies that the principal must obtain a randomized password from the password strength server. The password strength server strength-checks the password (enforces password policy).

- **pwd_mgmt_binding** – Specify a binding to the password strength server. This ERA uses the binding encoding type (*Section 4.5.2.5*).

4.5.2.1 Creating a Principal with Password Management ERAs

Use the **principal create** *principal* **-attribute {***attribute_list***}** command to create a principal and attach the **pwd_val_type** and **pwd_mgmt_binding** ERAs. For example:

```
dcecp> principal create fillmore -attribute {{pwd_val_type 2} \
{pwd_mgmt_binding \
{{dce /.:/pwd_strength pktprivacy secret name} \
{/.:/subsys/dce/pwd_mgmt/pwd_strength}}}}
```

4.5.2.2 Viewing a Principal's ERAs

Use the **principal show** *principal* **-xattrs** command to display a principal's ERAs. For example:

```
dcecp> principal show fillmore -xattrs
{pwd_val_type 2}
{pwd_mgmt_binding {{dce /.:/pwd_strength pktprivacy secret name}
{/.:/subsys/dce/pwd_mgmt/pwd_strength}}}
```

4.5.2.3 Adding Password Management ERAs to a Principal

Use the **principal modify** *principal* **-add {***attribute_list***}** command to attach the **pwd_val_type** and **pwd_mgmt_binding** ERAs to an existing principal. For example:

```
dcecp> principal modify buchanan -add {{pwd_val_type 2} \
{pwd_mgmt_binding \
{{dce /.:/pwd_strength pktprivacy secret name} \
{/.:/subsys/dce/pwd_mgmt/pwd_strength}}}}
```

4.5.2.4 Modifying a Principal’s Password Management ERA Values

Use the **principal modify** *principal* **-change {***attribute_list***}** command to change **pwd_val_type** or **pwd_mgmt_binding** values. For example:

```
dcecp> principal modify buchanan -change {pwd_val_type 3}
```

4.5.2.5 Using The Binding Encoding Type

The **pwd_mgmt_binding** ERA uses the **binding** encoding type:

{{*auth_info***} {***binding_info***}}**

which expands to:

{{*auth_serv_type name prot_level authentication_service authorization_service***} {***binding_info***}}**

Table 4-2 defines the parameters.

Table 4-2: Parameter Definitions for the Binding Encoding Type

Parameter	Values
auth_serv_type	Specifies the authentication type: ■ **none –** No authentication If you select **none**, specify no additional parameters except *binding_info*. ■ **dce –** Standard DCE authentication
name	Principal name of the server, usually **/.:/pwd_strength**.
prot_level	Protection level that determines the degree to which authenticated communications between the client and the server are protected by the authentication service: ■ **default –** Default protection level of **pkt**. ■ **none –** No authentication: no tickets are exchanged, no session keys are established, no client EPACs or names are certified, transmissions are in the clear. ■ **connect –** Authenticates only when client establishes a relationship with server. ■ **call –** Authenticates only at the beginning of each RPC when server receives the request. This level does not apply to RPCs made over a connection-based protocol sequence (that is, **ncacn_ip_tcp**). If this level is specified and the binding handle uses a connection-based protocol sequence, the routine uses the **pkt** protection level instead. *(prot_level continued next page)*

Table 4-2: Parameter Definitions for the Binding Encoding Type (Continued)

Parameter	Values
prot_level (continued)	■ **pkt –** Ensures that all data received is from the expected client. ■ **pktinteg –** Ensures and verifies that none of the data transferred between client and server has been modified. This is the highest protection level that is guaranteed to be present in the RPC runtime. ■ **pktprivacy –** Authenticates as specified by all of the previous levels and also encrypts each RPC argument value. This is the highest protection level, but it is not guaranteed to be present in the RPC runtime.
authentication_service	Specifies the authentication service. The exact level of protection provided by the authentication service is specified by *prot_level*. ■ **default –** DCE shared-secret key. ■ **none –** No authentication: no tickets are exchanged, no session keys established, client EPACs or names are not transmitted, and transmissions are in the clear. ■ **secret –** DCE shared-secret key authentication.
authorization_service	Specifies the authorization service. The validity and trustworthiness of authorization data, like any application data, is dependent on the authentication service and protection level specified. The supported authorization services are as follows: ■ **none –** Server performs no authorization. ■ **name –** Server performs authorization based on the client principal name. This value cannot be used if the authentication service is **none**. ■ **dce –** Server performs authorization by using the client's DCE EPAC sent to the server with each remote procedure call made with this binding. Generally, access is checked against DCE ACLs.
binding_info	Specifies the binding, which can be a string binding, a server entry name, or a list containing one or more string bindings or server entry names. The following example shows a server entry name binding: **/.:/hosts/host_name/dce_entity_name** The following example shows a string binding in standard syntax: **ncadg_udp_ip:130.105.96.3[1234]**

4.5.3 Running the Password Strength Server from the Command Line

If you wish, once the password strength server is configured (*Section 4.5.1 on page 56*), you can run the password strength server from the command line. You might wish to do this to override password policy for those principals that are configured to bind to the password strength server.

If you run the password strength server from the command line, this instance supersedes the instance started by the PC-DCE service. After you end the command line instance, the PC-DCE Service Panel will still show the password strength daemon as running, but its bindings will be unavailable.

pwd_strengthd.exe is located in *install_directory***bin**. For command line arguments, refer to *Section 4.5.3.1*.

4.5.3.1 Overriding Policy from the Command Line

By default, the password strength server enforces standard password policy (*Section 4.4 on page 54*). However, if you run the password strength server from the command line, you can use command line arguments to specify a separate password policy to be used for principals configured to use this server. In addition you can control a few other server options. For descriptions of all command line options, refer to *Table 4-3*.

If you don't specify an argument to explicitly override policy, the password strength server enforces policy. Be aware that if you specify options that are weaker than policy, the weaker options are used.

Table 4-3: Command Line Arguments for pwd_strengthd.exe

Option	Description
+all	Allows passwords to be all spaces.
-all	Prevents passwords from being all spaces.
+alp	Allows passwords to consist only of alphanumeric characters.
-alp	Prevents passwords from consisting only of alphanumeric characters.
-m *pwd_min_len*	Specifies the minimum length of the password.
-d	Runs **pwd_strengthd** in the foreground, which causes log messages to standard output.
-v	Runs in verbose mode.

For example, the following command runs the password strength server with arguments that prevent passwords from being all spaces, and sets the minimum password length to 6:

```
C:\>dce_login cell_admin mxydkwl
```

```
C:\>pwd_strengthd -all -m 6 -v -d
pwd_strengthd 05/06/2000 09:18:29 - Registering rsec_pwd_mgmt interface
pwd_strengthd 05/06/2000 09:18:31 - Setting up server context
pwd_strengthd 05/06/2000 09:18:33 - Listening on rsec_pwd_mgmt interface
```

4.5.4 Generating Randomized Passwords

A principal configured to use the password strength server with a **pwd_val_type** ERA having a value of 2 or 3 can (or will be required to) obtain a randomized password when creating or modifying its password. The following example shows how to use dcecp to obtain the password from the password strength server:

```
dcecp> set p [account generate delilah]
newgenpwd
```

This command requests a generated password from the password management server, places the new password in the *p* variable, and prints it to the screen (*newgenpwd*). Be sure to remember the new password.

Next, pass the value stored in *p* as the value of new password in an **account modify** or **account create** command:

```
dcecp> account modify delilah -password $p -mycurrentpwd -dce-
```

Never execute the following **dcecp** command, since the password will be changed in the account, but the user will not see the new password:

```
dcecp> acct mod delilah -password [acct gen delilah] -mycurrentpwd -dce-
```

4.5.4.1 Password Generation Restriction

The password strength server generates passwords according to registry policy only. It ignores the organization policy and any policy you specify using command line switches. For example, if a principal's organization's policy requires passwords to use non-alphanumeric characters, but registry policy does not, the password strength server will generate passwords that use alphanumeric characters only. To work around this restriction, you may need to temporarily change registry policy (*Section 4.4 on page 54*) when creating or modifying passwords.

4.5.5 Viewing Password Strength Server Log Messages

The log file for the sample password management server resides in *install_directory***dcelocal\var\security\pwd_strengthd.log**.

If you run the password strength server from the command line (*Section 4.5.3*), you can supply the **-d** command line argument to run the process in the foreground, which causes log messages to be directed to standard output.

4.6 Managing Password Expiration

You can cause passwords to expire by setting the following registry and organization policy attributes:

- **pwdlife** – passwords expire this number of seconds from the account creation time or the last time the password was changed.
- **pwdexpdate** – time at which all passwords in this domain (registry or organization) expire.

4.6.1 Viewing Password Expiration Settings

To view policy settings, including password expiration, for the entire registry, use the **registry show** *registry* **-policies** command. For example:

```
dcecp> registry show /.:/subsys/dce/sec/master -policies
{acctlife unlimited}
{maxtktlife +1-00:00:00.000I-----}
{maxtktrenew +28-00:00:00.000I-----}
{pwdalpha yes}
{pwdexpdate none}
```

```
{pwdlife unlimited}
{pwdminlen 6}
{pwdspaces yes}
```

To view policy settings for an organization, use the **organization show** *organization* **-policies** command. For example:

```
dcecp> org show zephyr -policies
{acctlife unlimited}
{pwdalpha no}
{pwdexpdate none}
{pwdlife unlimited}
{pwdminlen 0}
{pwdspaces yes}
```

4.6.2 Changing Password Expiration Settings

To change a policy attribute in the registry, use the dcecp **registry modify** command. For example, the following command changes the pwdlife to 7862400 (3 months) for the entire registry:

```
dcecp> registry modify /.:/subsys/dce/sec/master -change {pwdlife 7862400}
```

To change a policy attribute in an organization, use the dcecp **organization modify** command. For example, the following command sets the pwdexpdate to June 30, 2001 for organization zephyr:

```
dcecp> org modify zephyr -change {pwdexpdate 2001-06-30-00:00:00}
```

4.6.3 Overriding Password Expiration

By default, the security server disables logins for principals whose passwords have expired. There may be cases where you would prefer this not to happen; for instance, you don't want **cell_admin** to be locked out of the cell because of an expired password.

You can manage password expiration checking for a given principal by attaching an instance of the *passwd_override* ERA to the principal and specifying one of the following values:

0 (NONE) – Specifies that password expiration checking for the principal should not be overridden (that is, the principal should not be permitted to log in with an expired password.) Specifying **0 (NONE)** is equivalent to not attaching an ERA instance to the principal.

1 (OVERRIDE) – Specifies that password expiration checking for the principal should be overridden (that is, the principal should be permitted to log in with an expired password).

NOTE: In OSF DCE Version 1.2.1, the DCE privilege server removes ERAs from credentials requested by foreign cells. As a result, the **passwd_override** ERA has no effect for logins from foreign cells.

4.6.3.1 Creating a Principal with the passwd_override ERA

Use the **principal create** *principal* **-attribute {***attribute_list***}** command to create a principal and attach the **passwd_override** ERA. For example:

```
dcecp> principal create fillmore -attribute {passwd_override 1}
```

4.6.3.2 Viewing a Principal's passwd_override ERA

Use the **principal show** *principal* **-xattrs** command to display a principal's ERAs. For example:

```
dcecp> principal show fillmore -xattrs
{passwd_override 1}
```

4.6.3.3 Adding the passwd_override ERA to a Principal

Use the **principal modify** *principal* **-add {***attribute_list***}** command to attach the **passwd_override** ERA to an existing principal. For example:

```
dcecp> principal modify cell_admin -add {passwd_override 1}
```

4.6.3.4 Modifying a Principal's passwd_override ERA Value

Use the **principal modify** *principal* **-change {***attribute_list***}** command to change the **passwd_override** value. For example:

```
dcecp> principal modify fillmore -change {passwd_override 0}
```

4.7 Integrated Login

Integrated login is a feature of PC-DCE that automatically logs users into DCE whenever they log into Windows. When the user logs out of Windows, integrated login logs the user out of DCE. Additionally, if a user changes his Windows password, the integrated login feature automatically changes the DCE password.

Integrated login is a popular feature on client systems because it provides single sign-on capability, because it enables the user's DCE credentials to be available as soon as the system is done booting, and because it saves the user a login step.

For integrated login to work, it must be enabled through the PC-DCE Control Panel configuration program, and the user's Windows and DCE user names and passwords must be identical, including case. For instructions on configuring integrated login, refer to the PC-DCE Configuration Panel online help system.

You can verify that integrated login is working by logging into Windows and then issuing the **klist** command from a command prompt. The **klist** command displays the currently logged in principal name, which should match the Windows user name.

The integrated login function creates a global login context for the user. For more information on login contexts, refer to *Section 4.8*.

4.8 Login Contexts

A login context is the set of security credentials associated with a user logged into DCE. In UNIX, when a user creates a new process, that process normally inherits the user’s login context through an environment variable. However, Windows treats new processes differently than UNIX does. As a consequence, login contexts created in the normal fashion (through the DCE **sec_login** APIs) are per-process login contexts, and are not inherited by new processes.

PC-DCE includes a feature called the default login context that provides behavior similar to that seen on UNIX. When a user logs into DCE using either integrated login (*Section 4.7*) or the **dce_login** command, PC-DCE creates a default login context and stores it in the registry. Any new processes created by the user inherit the default login context.

CHAPTER 5

Managing the PC-DCE Cell and Servers

This chapter describes how to manage PC-DCE server replicas and how to backup PC-DCE server and configuration data. This chapter contains the following sections:

5.1 Definitions

Because DCE terminology regarding replicas differs somewhat as it applies to CDS versus the Security Service, it can sometimes lead to confusion. The following definitions clarify DCE replica terminology.

Replica – For CDS, a replica is a copy of a CDS directory.

For the Security Service, a replica is a complete copy of the security registry database.

Master – For CDS, a master is a writeable copy of a CDS directory. There can be only one master copy of a directory.

For the Security Service, the master is the writeable copy of the security registry. There can be only one master.

Slave – Read-only copy of the security registry database.

Primary Server – PC-DCE's term for a CDS server that contains, by default, a clearinghouse that contains master copies of all CDS directories. Although it is possible for master directories to be distributed among multiple clearinghouses, the default (and typical) scenario is for one clearinghouse to contain all the master directories.

Replica Server – For CDS, replica server is PC-DCE's term for a CDS server that contains readonly copies of CDS directories. By default, a PC-DCE replica server contains a clearinghouse that contains a readonly copy of the root directory (**/.:**) but copies of no other directories. A default CDS replica server is not a backup server.

For the Security Service, a replica server is a server that maintains a slave copy of the registry.

Backup Server – For CDS, a backup server is a replica server that contains readonly copies of all CDS directories.

For the Security Service, a backup server is a replica server.

5.2 Server Locations and Requirements

This section describes location, hardware, and security requirements for primary and replica servers.

5.2.1 Locating Servers

As part of your topology design you must decide where to locate your CDS and security servers.

5.2.1.1 For Performance

Because certain full client requests can only be fulfilled by the CDS server that owns the clearinghouse containing the master **hosts** directory replica, you should locate the master replica for this directory as close as possible to the majority of your users. Then locate additional replicas in remote locations to improve performance in those locations.

You can further tune your CDS database by locating the master directory for each host (**/.:/hosts/***hostname*) in a clearinghouse that is local to that host. When you initially configure a client system, the PC-DCE configuration program creates the host's master directory in the clearinghouse that contains the master **hosts** directory, even if that clearinghouse is not local. So, to perform this bit of fine-tuning, you may need to re-assign mastership for many host replicas.

Situate servers in the network topology so as to avoid any bottlenecks due to slow network links.

5.2.1.2 For Fault-Tolerance

If your cell serves more than a focused geographic area such as a building or small campus, you should locate your primary and backup servers in different locations. In case of primary server downtime due to power outage or other local catastrophe, the backup server will not be affected.

5.2.1.3 For Security

Make sure that all primary and replica servers, especially the master security server, are located in a physically secure area with controlled physical and login access.

5.3 Managing Cells

This section describes how to obtain information about a cell, such as the cell's DCE servers and hosts, and whether the cell's services and clients are working.

NOTE: You cannot use the **cell backup** command to back up PC-DCE servers. For instructions on backing up PC-DCE servers, refer to *Section 5.8 on page 82*.

5.3.1 Showing All Configured DCE Servers and DCE Hosts

To show configured DCE servers and hosts in a cell, enter the **dcecp cell show** command. This command returns a list of servers grouped by type, along with a list of DCE hosts, as follows:

- **secservers** – Each value is the name of a security server.
- **cdsservers** – Each value is the name of a machine running a CDS server. The name is the simple name found under **/.:/hosts**. A clearinghouse must be configured on that machine.
- **dtsservers** – Each value is the name of a DTS server in the cell.
- **hosts** – Each value is the name of a host in the cell, including machines mentioned previously as servers.

The following example shows the names of all the configured DCE servers and hosts in the local cell:

```
dcecp> cell show
{secservers
 /.../mycell.goodco.com/subsys/dce/sec/master}
{cdsservers
 /.../mycell.goodco.com/hosts/darwin}
{dtsservers
 /.../mycell.goodco.com/hosts/banks}
{hosts
 /.../mycell.goodco.com/hosts/banks
 /.../mycell.goodco.com/hosts/cook
 /.../mycell.goodco.com/hosts/darwin
 /.../mycell.goodco.com/hosts/wilson}
```

If you have the necessary permission, you can show the configured DCE servers and hosts in another cell by including that cell's name as an argument as shown in the following example:

```
dcecp> cell show /.../othercell.goodco.com
{secservers
 /.../othercell.goodco.com/subsys/dce/sec/master}
{cdsservers
 /.../othercell.goodco.com/hosts/magpie}
{dtsservers
 /.../othercell.goodco.com/hosts/bluejay}
{hosts
 /.../othercell.goodco.com/hosts/bluejay
 /.../othercell.goodco.com/hosts/finch
 /.../othercell.goodco.com/hosts/magpie
 /.../othercell.goodco.com/hosts/redpoll}
```

5.3.2 Testing Whether a DCE Server is Running

You can use the **server ping** command to test whether a server process is running. The **server ping** command returns a **1** if it the server is accessible and a **0** if it is not accessible.

The following example tests whether the master security server is accessible on the network:

```
dcecp> server ping /.:/subsys/dce/sec/master
1
```

Use the **cell show** command to obtain a list of servers to input into the **server ping** command.

5.3.3 Testing Whether a DCE Host is Running

Because DCE communications often involve several steps before clients communicate with their servers, communication failures can be difficult to diagnose. For instance, a server may not be running on a host or the DCE services may not be currently running, even though the host has been configured into the cell.

You can use a **server ping** command to test whether a server process is running but, if this fails, you can use the **host ping** command to test whether a host's DCE services are accessible on the network. This command returns a **1** if it is accessible and a **0** if it is not. The **host ping** operation tests for the presence of the remote host's DCE daemon (**dced**).

The following example tests whether **dced** on host **cook** is accessible on the network:

```
dcecp> host ping /.:/hosts/cook
1
```

5.3.4 Testing Cell Operation

You can test whether a cell's DCE services are running by entering the **cell ping** command.

If called with no option, the **cell ping** command performs a **server ping** operation on the master security server, on the CDS server that has a master clearinghouse, and all the DTS servers in the cell. Use the **-replicas** option to test CDS and security service replicas as well as the masters. The **-clients** option tests every DCE host in the cell by looping though the **/.:/hosts** directory in CDS and performing a **host ping**, with each host name as an argument.

In case of failure, the operation generates an error and returns a list of servers or hosts that could not be contacted. For any successes, the operation returns the message **DCE services available**. For successes with the **-clients** option, the message is **DCE clients available**.

The following example pings the names of all the configured master DCE servers in the local cell:

```
dcecp> cell ping
DCE services available
```

The following example pings the names of all the configured DCE hosts in the local cell. Depending on the size of a cell and timeout values set, this command can take a long time (from several to many minutes) to complete.

```
dcecp> cell ping -clients
DCE clients available
```

If you have the necessary permission, you can ping the configured DCE servers and hosts in another cell by including that cell's name as an argument as shown in the following example:

```
dcecp> cell ping /.../othercell.goodco.com
DCE services available
```

5.4 Incorporating Replicas into a Cell

Because your users all rely on the CDS and the security registry, you should take steps to ensure that if a primary server becomes unavailable, an adequate backup is available. DCE replica functionality lets you create backup servers that continue to provide cell directory and security services if the primary servers are unavailable.

5.4.1 Choosing the Number of Replicas

Your cell should contain, in addition to the primary server for CDS, at least one backup server. A backup server is a replica server that contains read-only replicas of all CDS directories, rather than just the root, which is the default configuration for a CDS replica server. It should be standard operating procedure to keep the CDS directories maintained on backup servers complete and up-to-date.

Your cell should also contain, in addition to the master security server, at least one slave server. A security server slave may also be referred to as a replica server or a backup security server.

In environments that cannot tolerate downtime, it is prudent to maintain at least two full CDS replica servers and two security slaves. Then if a primary server becomes unavailable, you can configure one backup as the new primary server and still have a backup already in place.

5.5 Managing CDS Replicas

This section describes how to manage CDS replicas:

5.5.1 Understanding the PC-DCE Default CDS Replica Configuration

When you configure a cell using PC-DCE servers, you configure the CDS server to either be a *primary server* or a *replica server*. The primary server maintains a clearinghouse that contains the master replicas of all CDS directories. A replica server, by default, maintains a clearinghouse that contains a readonly replica of the root directory (**/.:**) only.

5.5.2 Why Modify the Default PC-DCE CDS Replica Configuration?

You can use DCE command line tools to create replicas of additional directories in the clearinghouse maintained by a replica server. You can also change the mastership of specific directories.

Why would you want to do any of this? Here are several possible reasons:

- You want to increase the ability of replica servers to perform load sharing. In this case you would want to replicate additional directories on replica servers so they can serve information from these directories.
- You want to assign a replica server mastership of certain directories in order to enhance performance in geographically-dispersed cell. For example, when a full DCE client initializes it exports objects to the master directory for its host (**/.:/hosts/***hostname*). You may wish to move the mastership of certain hosts subdirectories to the replica server closest to the corresponding hosts.
- You want to convert an existing replica server to be the primary server. You may want to do this for performance reasons or if the primary server has experienced a hardware failure.
- You want to ensure that clients will be able to "failover" if the master is down (see *Appendix B on page 123*).

The following sections describe how to make these modifications to the default CDS replica structure.

5.5.3 Basic CDS Replica Management Tasks

This section describes basic CDS replica management tasks. A later section describes how to apply these basic tasks to more complex management procedures.

5.5.3.1 Determining the Structure of the CDS Directory Tree

The first thing you should do if you are planning to add or delete replicas or change replica mastership is to familiarize yourself with the structure of the current CDS directory tree. To do so, you explore the tree using the following **dcecp** command:

directory list -directories *directory*

This command lists all of the directories subordinate to *directory*. By starting at the root directory (**/.:**) and working your way downward, you can discover all of the directories. For example:

```
C:\> dce_login cell_admin -dce-
C:\> dcecp
dcecp> directory list -directories /.:
/.../longwood/hosts /.../longwood/subsys /.../longwood/users
```

This shows the three directories under the root. Now we will explore each of these subdirectories:

```
dcecp> directory list -directories /.:/hosts
/.../longwood/hosts/banks.explorers.com
/.../longwood/hosts/darwin.explorers.com
```

```
dcecp> directory list -directories /.:/hosts/banks.explorers.com
dcecp> directory list -directories /.:/hosts/darwin.explorers.com
dcecp>
```

There are no directories under host directories.

```
dcecp> directory list -directories /.:/subsys
/.../longwood/subsys/dce
dcecp> directory list -directories /.:/subsys/dce
/.../longwood/subsys/dce/sec
dcecp> directory list -directories /.:/subsys/dce/sec
dcecp>
```

There are no directories under **subsys/dce/sec**.

```
dcecp> directory list -directories /.:/users
/.../longwood/users/joseph /.../longwood/users/charles
```

```
dcecp> directory list -directories /.:/users/bob
dcecp> directory list -directories /.:/users/charles
dcecp>
```

There are no directories under user directories.

The structure of this CDS directory tree is therefore:

```
/.:
```

```
/.:/hosts
/.:/hosts/banks.explorers.com
/.:/hosts/darwin.explorers.com
```

```
/.:/subsys
/.:/subsys/dce
/.:/subsys/dce/sec

/.:/users
/.:/users/joseph
/.:/users/charles
```

5.5.3.2 Creating a New Directory Replica

To create replicas of CDS directories, use the following dcecp command:

directory create *directory* **-replica -clearinghouse** *clearinghouse*

For example, if you want to create a replica of **/.:/hosts** in a clearinghouse **darwin_ch**, you would enter:

```
C:\> dce_login cell_admin -dce-
C:\> dcecp
dcecp> directory create /.:/hosts -replica -clearinghouse /.:/darwin_ch
```

After you create the directory replica, you synchronize its contents with the contents of the corresponding master directory (*Section 5.5.3.3*).

NOTE: To create replicas of all directories below (and including) the root directory, use the Full Replica option on the Custom tab of the PC-DCE Configuration Panel.

You can verify that all of the replicas have been created by entering the **clearinghouse show** command and checking the `Dir_Name` entries. For example:

```
dcecp> clearinghouse show /.:/shadow
{CDS_CTS 1999-05-11-18:11:21.697000100/00-10-4b-9a-fb-74}
{CDS_UTS 1999-05-11-18:11:25.753000500/00-10-4b-9a-fb-74}
{CDS_ObjectUUID 0065c6b0-72c9-1738-9286-00104b9afb74}
{CDS_AllUpTo 1999-05-11-18:11:24.200000100/00-10-4b-9a-fb-74}
{CDS_DirectoryVersion 4.0}
{CDS_CHName /.../longwood/shadow}
{CDS_CHLastAddress
 {Tower {ncacn_ip_tcp 190.82.110.235}}
 {Tower {ncadg_ip_udp 190.82.110.235}}}
{CDS_CHState on}
{CDS_CHDirectories
{{Dir_UUID 0006ddd1-ff50-16d2-9270-00104b9afb74}
 {Dir_Name /.../longwood}}
{{Dir_UUID 007779f0-ff55-16d2-9270-00104b9afb74}
 {Dir_Name /.../longwood/hosts}}
{{Dir_UUID 00609690-ff55-16d2-9270-00104b9afb74}
 {Dir_Name /.../longwood/subsys}}
{{Dir_UUID 00534690-ff55-16d2-9270-00104b9afb74}
 {Dir_Name /.../longwood/users}}
{{Dir_UUID 00962580-ff9b-16d2-9270-00104b9afb74}
 {Dir_Name /.../longwood/hosts/banks.explorers.com}}
{{Dir_UUID 007f1b10-ff55-16d2-9270-00104b9afb74}
 {Dir_Name /.../longwood/hosts/darwin.explorers.com}}
{{Dir_UUID 006ccb90-ff55-16d2-9270-00104b9afb74}
 {Dir_Name /.../longwood/subsys/dce}}
```

```
{{Dir_UUID 0072e610-ff55-16d2-9270-00104b9afb74}
 {Dir_Name /.../longwood/subsys/dce/sec}}
{{Dir_UUID 006ce530-ff55-16d2-9270-00104b9afb74}
 {Dir_Name /.../longwood/users/joseph}}
{{Dir_UUID 00923620-ff55-16d2-9270-00104b9afb74}
 {Dir_Name /.../longwood/users/charles}}
{CDS_ReplicaVersion 4.0}
{CDS_NSCellname /.../longwood}
```

5.5.3.3 Synchronizing a Directory Replica

To synchronize a directory means to update a read-only replica directory with the contents of the corresponding master directory. This normally happens on a periodic basis (default: every hour).

Using the **cdsreplica create** command to create replicas of all directories below (and including) the root directory automatically synchronizes all replica directories as it creates them.

To synchronize a directory immediately, use the following **dcecp** command:

directory synchronize *directory*

For example, to synchronize the contents of a newly created replica of the **/.:/hosts** directory with the contents of the corresponding master directory:

```
dcecp> directory synchronize /.:/hosts
```

5.5.3.4 Changing Mastership of a Replica

If you want to change the mastership of a replica, you use the following **cdscp** command:

set directory *directory* **to new epoch master**
new_clearinghouse **readonly** *old_clearinghouse*

This command sets the old master directory to be a read–only replica. You would do this if you want to keep the old primary server on the network as a replica server.

If you want to eliminate the old master directory, use the **exclude** option:

set directory *directory* **to new epoch master**
new_clearinghouse **exclude** *old_clearinghouse*

For example, you may want to eliminate directories if you are removing the old master server from the network.

The following command changes the mastership of the directory **/.:/hosts/darwin.explorers.com** to the **darwin_ch** clearinghouse. The directory becomes readonly on the old master clearinghouse **banks_ch**:

```
C:\> dce_login cell_admin -dce-
C:\> cdscp
cdscp> set directory /.:/hosts/darwin.explorers.com to new epoch master /.:/
darwin_ch readonly /.:/banks_ch
```

5.5.4 CDS Replica Management Procedures

This section describes how to apply basic CDS replica management tasks (*Section 5.5.3 on page 71*) to more complex replica management procedures.

5.5.4.1 Creating a Backup Server

If you want to create a backup server that provides a full backup of all CDS directories if the primary CDS server experiences data loss, you must determine the structure of the CDS namespace (*Section 5.5.3.1 on page 71*) and create replicas of all directories on your backup server (*Section 5.5.3.2 on page 72*).

You must make it standard operating procedure to keep the backup server current. Directories that exist on both master and read-only replicas are automatically synchronized on a periodic basis. However, if someone creates a new directory on the master, the new directory is not automatically replicated.

5.5.4.2 Creating a Backup Server for Failover

If you want to create a backup server that will simply allow clients to start up if the primary server is temporarily unavailable, at a minimum you must replicate the **/.:/subsys/dce/sec** directory, which stores the locations of security servers. For example:

```
dcecp> directory create /.:/subsys
dcecp> directory create /.:/subsys/dce
dcecp> directory create /.:/subsys/dce/sec
dcecp> directory synch /.:/subsys/dce/sec
dcecp> directory synch /.:/subsys/dce
dcecp> directory synch /.:/subsys
dcecp> directory synch /.:
```

5.5.4.3 Converting a Backup Server to a Primary Server

The following procedure is a general overview. Details on each step follow the general overview.

1 Determine the structure of the CDS directory tree.

2 If the replica server clearinghouse does not contain a complete set of directory replicas, create new directory replicas.

> NOTE: If the primary server is unavailable, you will need to restore the CDS directories from backups (*Section 5.8 on page 82*).

3 Change mastership of the directory replicas.

4 Verify changes.

To determine the structure of the CDS directory tree:

Use the procedure described in *Section 5.5.3.1 on page 71*.

To create new directory replicas:

Creating new directory replicas is described in *Section 5.5.3.2 on page 72*. The following is an example of the commands you would issue to create replicas on a replica server whose clearinghouse contains a copy of the root directory only. This example creates the replicas in the clearinghouse **/.:/darwin_ch**:

```
C:\> dce_login cell_admin -dce-
C:\> dcecp
dcecp> directory create /.:/hosts -replica -clearinghouse /.:/darwin_ch
dcecp> directory create /.:/hosts/banks.entegrity.com -replica
-clearinghouse /.:/darwin_ch
dcecp> directory create /.:/hosts/darwin.entegrity.com -replica
-clearinghouse /.:/darwin_ch
dcecp> directory create /.:/subsys -replica -clearinghouse /.:/darwin_ch
dcecp> directory create /.:/subsys/dce -replica -clearinghouse /.:/darwin_ch
dcecp> directory create /.:/subsys/dce/sec -replica -clearinghouse /.:/
darwin_ch
dcecp> directory create /.:/users -replica -clearinghouse /.:/darwin_ch
dcecp> directory create /.:/users/joseph -replica -clearinghouse /.:/
darwin_ch
dcecp> directory create /.:/users/charles -replica -clearinghouse /.:/
darwin_ch
```

If the primary server is available, synchronize the directories. Notice that you synchronize starting with the ends of the branches (the leaves) and working up to the root.

```
dcecp> directory synchronize /.:/users/joseph
dcecp> directory synchronize /.:/users/charles
dcecp> directory synchronize /.:/users
dcecp> directory synchronize /.:/subsys/dce/sec
dcecp> directory synchronize /.:/subsys/dce
dcecp> directory synchronize /.:/subsys
dcecp> directory synchronize /.:/hosts/darwin.entegrity.com
dcecp> directory synchronize /.:/hosts/banks.entegrity.com
dcecp> directory synchronize /.:/hosts
dcecp> directory synchronize /.:
```

To change mastership of the directory replicas:

If the old primary server is to be converted to a replica server, issue the **dcecp directory modify** command with the **readonly** option. In the following example, the clearinghouse for the old primary server is **/.:/banks_ch**.

```
dcecp> directory modify /.: -master /.:/darwin_ch -readonly /.:/banks_ch
dcecp> directory modify /.:/hosts -master /.:/darwin_ch
-readonly /.:/banks_ch
dcecp> directory modify /.:/hosts/banks.entegrity.com -master /.:/darwin_ch
-readonly /.:/banks_ch
dcecp> directory modify /.:/hosts/darwin.entegrity.com -master /.:/darwin_ch
-readonly /.:/banks_ch
dcecp> directory modify /.:/subsys -master /.:/darwin_ch
-readonly /.:/banks_ch
```

```
dcecp> directory modify /.:/subsys/dce -master /.:/darwin_ch
-readonly /.:/banks_ch
dcecp> directory modify /.:/subsys/dce/sec -master /.:/darwin_ch
-readonly /.:/banks_ch
dcecp> directory modify /.:/users -master /.:/darwin_ch
-readonly /.:/banks_ch
dcecp> directory modify /.:/users/joseph -master /.:/darwin_ch
-readonly /.:/banks_ch
dcecp> directory modify /.:/users/charles -master /.:/darwin_ch
-readonly /.:/banks_ch
```

If the old primary server will no longer be in service, issue the above commands using the **exclude** option in place of the **readonly** option. For example:

```
dcecp> directory modify /.: -master /.:/darwin_ch -exclude /.:/banks_ch
```

NOTE: If you are removing the old primary server from service, you must remove all references to it from the cell as described in *Section 5.7 on page 79*.

To verify your changes:

To verify that mastership has changed, issue the **dcecp** command **directory show** on at least one directory. For example, the following command displays information on the root (**/.:**) directory. In the example command output, the arrows (<-----) point to the pertinent lines.

```
dcecp> directory show /.:
{RPC_ClassVersion {01 00}}
{CDS_CTS 2000-04-02-14:53:41.113000100/00-60-08-a1-4e-cf}
{CDS_UTS 2000-04-02-19:13:51.127000700/00-60-08-a1-4e-cf}
{CDS_ObjectUUID 00113e11-a675-1523-ba3f-006008a14ecf}
{CDS_Replicas
 {{CH_UUID 005a06e0-b994-1523-baa9-006008a14ecf}
  {CH_Name /.../longwood/darwin_ch}                          <-----
  {Replica_Type Master}                                      <-----
  {Tower {ncacn_ip_tcp 192.92.110.141}}
  {Tower {ncadg_ip_udp 192.92.110.141}}}
 {{CH_UUID 0004baf0-a72d-1523-9616-00802964ff95}
  {CH_Name /.../longwood/banks_ch}                           <-----
  {Replica_Type ReadOnly}                                    <-----
  {Tower {ncacn_ip_tcp 192.92.110.53}}
  {Tower {ncadg_ip_udp 192.92.110.53}}}}
{CDS_AllUpTo 2000-04-02-19:13:51.497000100/00-60-08-a1-4e-cf}
{CDS_Convergence medium}
{CDS_InCHName new_dir}
{CDS_DirectoryVersion 4.0}
{CDS_ReplicaState on}
{CDS_ReplicaType ReadOnly}
{CDS_LastSkulk 2000-04-02-19:13:51.497000100/00-60-08-a1-4e-cf}
{CDS_LastUpdate 2000-04-02-19:10:43.964000100/00-80-29-64-ff-95}
{CDS_Epoch 001360f0-e36f-1523-baa9-006008a14ecf}
{CDS_ReplicaVersion 4.0}
```

5.6 Managing Security Service Replicas

You use the PC-DCE Configuration Panel to specify whether the security server you are configuring is to be a primary server (master) or a replica server (slave). The first server you create for a cell must be the master, and then it is easy to configure additional slaves.

You use the DCE command line tools to promote a slave to a master or demote a master to a slave. You may need to do this if the master experiences a hardware failure or if you want to run the master on a faster machine.

The following sections describe how to perform these procedures.

5.6.1 Promoting a Slave to Master

This section describes how to promote a slave to a master. In this scenario, the current master is available and will be converted to a slave.

The following procedure is a general overview. Details on each step follow the general overview.

1. List the security servers to obtain the correct names.
2. Designate the new master and slave.
3. Verify that mastership has changed.
4. Verify that the new slave is correctly updating its copy of the registry.

To list the security servers to obtain the correct names:

Use the **dcecp** command **registry catalog**. For example:

```
C:\> dce_login cell_admin -dce-
C:\> dcecp
dcecp> registry catalog
/.../longwood/subsys/dce/sec/banks.explorers.com
/.../longwood/subsys/dce/sec/darwin.explorers.com
```

To designate the new master and slave:

Use the **dcecp** command **registry designate**. The following example designates **darwin.explorers.com**, currently a slave, to become the master, and **banks.explorers.com**, currently the master, to become a slave:

```
dcecp> registry designate /.:/subsys/dce/sec/banks.explorers.com -slave
dcecp> registry designate /.:/subsys/dce/sec/darwin.explorers.com -master
```

To verify that mastership has changed:

Use the **sec_admin** command **lrep -all** to verify that the mastership has changed. In the following example command output, the arrows (<-----) point to the pertinent lines.

```
dcecp> sec_admin
Default replica:  /.../longwood/subsys/dce/sec/darwin.explorers.com
Default cell:     /.../longwood
sec_admin> lrep -all
Default replica:  /.../longwood/subsys/dce/sec/darwin.explorers.com
Default cell:     /.../longwood

subsys/dce/sec/darwin.explorers.com (master)
          Instance id: 002eff42-b985-1523-b204-006008a14ecf
          Addresses:               ncacn_ip_tcp:191.92.110.141[]
                                   ncadg_ip_udp:191.92.110.141[]
          State:                   in service - master               <----
          Last update received at: Thu Apr 02 14:45:31 2000
          Last update's seqno:     0.235

subsys/dce/sec/banks.explorers.com
          Instance id: 00881bc0-a718-1523-a4f9-00802964ff95
          Addresses:               ncacn_ip_tcp:191.92.110.53[]
                                   ncadg_ip_udp:191.92.110.53[]
          State:                   in service - slave                <----
          Last update received at: Thu Apr 02 14:45:31 2000
          Last update's seqno:     0.235
          Propagation state:       ready for updates
          Last update delivered:   Thu Apr 02 14:45:31 2000
          Last update's seqno:     0.235
          Number of outstanding updates: 0
          Last comm status:        Successful completion (dce / svc)
```

To verify that the new slave is correctly updating its copy of the registry:

Change the registry by creating a new user, and then issue another **lrep -all** command to determine whether the seqno values have changed. In the following example command output, the arrows (<-----) point to the pertinent lines.

```
dcecp> user create wendy -group none -org none -password wendy -mypwd -dce-
dcecp> sec_admin
Default replica:  /.../longwood/subsys/dce/sec/darwin.explorers.com
Default cell:     /.../longwood
sec_admin> lrep -all
Default replica:  /.../longwood/subsys/dce/sec/darwin.explorers.com
Default cell:     /.../longwood

subsys/dce/sec/darwin.explorers.com (master)
          Instance id: 002eff42-b985-1523-b204-006008a14ecf
          Addresses:               ncacn_ip_tcp:192.92.110.141[]
                                   ncadg_ip_udp:192.92.110.141[]
          State:                   in service - master
          Last update received at: Thu Apr 02 15:07:48 2000
          Last update's seqno:     0.262                             <----

subsys/dce/sec/banks.explorers.com
          Instance id: 00881bc0-a718-1523-a4f9-00802964ff95
          Addresses:               ncacn_ip_tcp:192.92.110.53[]
                                   ncadg_ip_udp:192.92.110.53[]
          State:                   in service - slave
          Last update received at: Thu Apr 02 15:07:48 2000
          Last update's seqno:     0.262                             <----
          Propagation state:       ready for updates
          Last update delivered:   Thu Apr 02 15:07:48 2000
          Last update's seqno:     0.262
          Number of outstanding updates: 0
          Last comm status:        Successful completion (dce / svc)
```

If the slave's **last update's seqno** is lower than the master's **last update's seqno**, then the slave is not updating its copy of the registry. This typically occurs when the slave was the original master and is named **/.:/subsys/dce/sec/master**. This is a known problem in DCE. If this occurs, you will have to completely remove the slave from the cell and reconfigure it, as described in *Section 5.7*.

5.7 Removing and Reconfiguring a PC-DCE Server

This procedure describes how to remove and reconfigure a PC-DCE server running both Security and CDS servers.

5.7.1 Removing Cell References to the Server

If you are planning to reconfigure a server, but not reconfigure the cell, you must remove cell references to the server prior to reconfiguring.

If a machine is no longer serving as a replica due either to reconfiguration or system failure, it is necessary to remove references to the replica from the cell registry. Not doing this could cause time delays should the master go down and will cause failure if you try to configure the machine as a replica in the future.

The following procedure is a general overview. Details on each step follow the general overview.

1. Remove the old replica's machine and cds-server principals.
2. Remove the old replica from the **/.:/sec** rpcgroup.
3. Delete the old replica.
4. Exclude all CDS directories from the old CDS server's clearinghouse.
5. Delete the old clearinghouse object.
6. Delete the old master **hosts** directory and its remaining contents.
7. Remove the DTS server, if present.

To remove the old replica's machine and cds-server principals:

Use the **dcecp principal delete** command. For example:

```
C:\> dce_login cell_admin -dce-
dcecp> principal delete hosts/banks.entegrity.com/self
dcecp> principal delete hosts/banks.entegrity.com/cds-server
```

To remove the old replica from the /.:/sec rpcgroup:

Use the **dcecp** command **rcpgroup remove**. For example:

```
dcecp> rpcgroup remove /.:/sec -member /.:/subsys/dce/sec/master
```

To delete the old security replica:

Use the **dcecp** command **registry delete** command with the **-force** option to remove the replica. The following example removes a reference to an old master:

```
dcecp> registry delete subsys/dce/sec/master -force
dcecp>
```

To exclude all CDS directories from the old CDS server’s clearinghouse:

Use the **dcecp** command **directory modify** with the **exclude** option. For example:

```
C:\>dcecp
dcecp> directory modify /.: -master /.:/darwin_ch -exclude /.:/banks_ch
dcecp> directory modify /.:/hosts -master /.:/darwin_ch
-exclude /.:/banks_ch
dcecp> directory modify /.:/hosts/darwin.entegrity.com -master
/.:/darwin_ch -exclude /.:/banks_ch
dcecp> directory modify /.:/hosts/banks.entegrity.com -master
/.:/darwin_ch exclude /.:/banks_ch
dcecp> directory modify /.:/subsys -master /.:/darwin_ch
-exclude /.:/banks_ch
dcecp> directory modify /.:/subsys/dce -master /.:/darwin_ch
-exclude /.:/banks_ch
dcecp> directory modify /.:/subsys/dce/sec -master /.:/darwin_ch
-exclude /.:/banks_ch
```

You may see a message regarding a replica that does not exist at the clearinghouse you are excluding. This is harmless because if the replica is not there you do not have to exclude it.

To delete the old clearinghouse object:

Use the **dcecp** command **obj delete**. For example:

```
dcecp> obj delete /.:/banks_ch
```

To delete the old master hosts directory and its remaining contents:

Use the **dcecp** command **dir delete**. For example:

```
dcecp> dir delete /.:/hosts/banks.entegrity.com -tree
```

To remove the DTS server:

1 Use the **dcecp** command **rpcprofile show** to obtain the interface UUID of the dts-entity:

```
dcecp> rpcprofile show /.:/lan-profile
{{019ee420-682d-11c9-a607-08002b0dea7a 1.0} /.../longwood/hosts/
banks.entegrity.com/dts-entity 0 {Time Server entry}}
```

2 Use the **dcecp** command **rpcprofile remove** to remove the dts-entity:

```
dcecp> rpcprofile remove /.:/lan-profile
-member /.:/hosts/banks.entegrity.com/dts-entity
-interface {019ee420-682d-11c9-a607-08002b0dea7a 1.0}
```

3. If you are running a global time server, you must remove an additional entry. To find the UUID of the global dts-entity, use the following command. In this example, the UUID is in the last entry.

```
dcecp> rpcprofile show /.:/cell-profile
{{d46113d0-a848-11cb-b863-08001e046aa5 2.0} /.../longwood/sec 0 rs_bind}
{{0d7c1e50-113a-11ca-b71f-08001e01dc6c 1.0} /.../longwood/sec-v1 0
secidmap}
{{8f73de50-768c-11ca-bffc-08001e039431 1.0} /.../longwood/sec 0 krb5rpc}
{{b1e338f8-9533-11c9-a34a-08001e019c1e 1.1} /.../longwood/sec 0 rpriv}
{{b1e338f8-9533-11c9-a34a-08001e019c1e 1.0} /.../longwood/sec 0 rpriv}
{{6f264242-b9f8-11c9-ad31-08002b0dc035 1.0} /.../longwood/lan-profile 0
LAN}
{{4d37f2dd-ed43-0000-02c0-37cf2e000001 4.0} /.../longwood/fs 0 fs}
{{eb814e2a-0099-11ca-8678-02608c2ea96e 4.0} /.../longwood/subsys/dce/dfs/
bak 0 bak}
{{17579714-82c9-11c9-8a59-08002b0dc035 1.0} /.../longwood/hosts/
banks.entegrity.com/dts-entity 0 {Global Time Server}}
```

4. To remove the global dts-entity, use the **dcecp** command **rpcprofile remove**:

```
dcecp> rpcprofile remove /.:/cell-profile
-member /.:/hosts/banks.entegrity.com/dts-entity
-interface {17579714-82c9-11c9-8a59-08002b0dc035 1.0}
```

5.7.2 Reconfiguring the Server

If you are reconfiguring the server, perform the following additional procedure. Details on each step follow the general procedure.

1. Delete PC-DCE files that contain references to the server.
2. Clear the endpoint mapper.
3. Reconfigure the old master as a replica.
4. Verify that the new replica is updating its registry.

To delete PC-DCE files that contain references to the server:

On the old master, delete the following files:

- **PCDCE32\opt\dcelocal\dce_cf.db**
- **PCDCE32\opt\dcelocal\etc\security\pe_site**

To clear the endpoint mapper:

Reboot the machine to clear the Microsoft endpoint mapper.

To reconfigure an old master as a replica:

Use the PC-DCE Configuration Panel, accessed through the PC-DCE Service Panel (Configure button), to reconfigure the old master:

- On the Custom tab of the PC-DCE Configuration Panel, select *replica* for each server.
- Change the master server name to the name of the new master server.

If the configuration hangs on *Configuring clearinghouse*, perform a dce_login as cell_admin from the command line and wait a bit longer — this should fix the problem.

To verify the old master (new replica) is updating its registry:

Create a new user and then use the sec_admin **lrep -all** command to verify that the sequence numbers match. For details, refer to *Section 5.6.1 on page 77*.

5.8 Backing Up and Restoring a PC-DCE Server

In addition to having replicas in the cell for automatic failover, you should also back up your PC-DCE data. This section describes how to back up and restore PC-DCE server configuration and database data. These procedures apply to both the CDS and Security Service servers.

5.8.1 Backing Up (With Servers Stopped)

1. Stop PC-DCE on the system to be backed up:
   - Open the Windows control panel and open the PC-DCE Service Panel application.
   - Click the Stop DCE button.
2. Back up the registry key **HKEY_LOCAL_MACHINE/Software/Entegrity/DCE/Configuration**:
   - Start the registry editor (**regedt32.exe**).
   - Select the HKEY_LOCAL_MACHINE window, navigate to the Entegrity DCE configuration key, and select it by clicking on it.
   - Use the Registry > Save Key... command to make a file backup of the key.
3. Back up the following directories:
   - *install_directory***opt** (for example, **PCDCE32\opt**)
   - *install_directory***krb5** (for example, **PCDCE32\krb5**)
4. Restart PC-DCE from the PC-DCE Service Panel.

5.8.2 Backing Up (With Servers Running)

1 Put the master security server into maintenance mode:

```
C:\> dce_login cell_admin -dce-
C:\> sec_admin
default replica:  /.../longwood
Default cell:     /.../longwood
sec_admin> state -m
```

NOTE: The **state** command only works on the master. If **sec_admin** is not already bound to the master, it attempts to do so when you issue this command.

2 Disassociate the clearinghouse from the master CDS server. In the following example, **darwin** is the master:

```
C:\> dcecp
dcecp> clearinghouse disable /.:/darwin_ch
```

3 Back up the registry key **HKEY_LOCAL_MACHINE/Software/Entegrity/DCE/Configuration**:

- Start the registry editor (**regedt32.exe**).
- Select the HKEY_LOCAL_MACHINE window, navigate to the Entegrity DCE configuration key, and select it by clicking on it.
- Use the Registry > Save Key... command to make a file backup of the key.

4 Back up the following directories:

- *install_directory*\opt (for example, **PCDCE32\\opt**)
- *install_directory*\krb5 (for example, **PCDCE32\\krb5**)

5 Put the master security server into service mode:

```
sec_admin> state -s
```

6 Relocate the clearinghouse:

```
dcecp> clearinghouse create /.:/darwin_ch
```

5.8.3 Restoring

1 Stop PC-DCE on the system to be restored:

- Open the Windows control panel and open the PC-DCE Service Panel application.
- Click the Stop DCE button.

2 Restore the registry key **HKEY_LOCAL_MACHINE/Software/Entegrity/DCE/Configuration**:

- Start the registry editor (**regedt32.exe**).
- Select the HKEY_LOCAL_MACHINE window, navigate to the Entegrity DCE configuration key, and select it by clicking on it.
- Use the Registry > Restore... command to select the backup file and apply it to the registry.

3 Restore the following directories from the backups:

- *install_directory***opt**
- *install_directory***krb5**

4 Restart PC-DCE from the PC-DCE Service Panel.

5.8.4 PC-DCE Master Security Server Upgrade

If you are upgrading from PC-DCE 3.0 or any version before 2.2.1HF1, the master security server may require an upgrade providing an improved method for storing passwords in the security registry's account database.

If a PC-DCE security server is the master security server in your cell, *OR* if a PC-DCE replica security server has ever been promoted to be the master (even temporarily) *AND* the account registry database was changed while a PC-DCE security server was the master server, follow these steps:

1 From the PC-DCE Service Panel on the security server machine, click Stop DCE.

2 If there are any replicas configured in the cell, stop them by clicking Stop DCE on the PC-DCE Service Panel from the replica machines.

3 Back up the files located in:
install_directory\opt\dcelocal\var\security\rgy_data\

4 Back up the the file: install_directory\opt\dcelocal\var\security\.mkey.

5 From a command line, enter the following:

```
C:\> sec_salvage_db.exe –print

C:\>
```

6 From the command line, enter the following:

```
C:\> sec_salvage_db.exe –account
```

This upgrades the passwords in the account registry database on the PC-DCE master security server. The machine responds:

Will overwrite data at /opt/dcelocal/var/security/rgy_data/, do you wish to continue (y[es]) or abort sec_salvage_db.exe (n[o])?

7 Choose yes.

8 From the PC-DCE Service Panel on the master security server machine, click Start DCE to bring up the master security server.

9 From the PC-DCE Service Panel on the replica security server machine(s), click Start DCE to restart the replicas if any exist in the cell.

10 Login to DCE as cell_admin.

11 From a command line, run dcecp and perform a “registry sync” for each replica in the cell.

```
C:\> dcecp
dcecp> registry sync /.:/subsys/dce/sec/hostname
```

This upgrades the replica’s account databases.

5.9 Using Locksmith Mode to Repair Registry Damage

If the security registry is damaged by an intruder or other agent, and you are unable to log in as a cell administrator, you may be able to repair the damage by putting the security server into locksmith mode. Locksmith mode allows a special principal, the locksmith principal, to log in with special access privileges. As the locksmith principal, you can repair registry damage.

When you bring up a security server in locksmith mode, **secd** automatically creates a locksmith account or, if the locksmith account exists, it lets you supply a new password for that account.

5.9.1 Automatic Changes to the Locksmith Account

If the locksmith account exists when you start the security server in locksmith mode, the security server checks certain account and registry policy information and makes the changes shown in *Table 5-1* and *Table 5-2*. These changes ensure that you will be able to log into the locksmith account even if the account or registry policy was tampered with.

Table 5-1: Locksmith Account Changes Made by the Security Server

If the security server finds the...	It changes the....
Password-Valid Flag is set to **no**	Password-Valid Flag to **yes**
Account Expiration Date is set to less than the current time plus 1 hour	Account Expiration Date to the current time plus 1 hour
Client Flag is set to **no**	Client Flag to **yes**
Account-Valid Flag is set to **no**	Account-Valid Flag to **yes**
Good Since Date is set to greater than the current time	Good Since Date to the current time
Password Expiration Date is set to less than the current time plus 1 hour	Password Expiration Date to the current time plus 1 hour

Table 5-2: Registry Policy Changes Made by the Security Server

If the security server finds the...	It changes the....
Account Lifespan is set to less than the difference between the locksmith account creation date and the current time plus 1 hour	Account Lifespan to the current time plus 1 hour minus the locksmith account creation date
Password Expiration Date is set to greater than the time the password was last changed but less than the current time plus 1 hour	Password Expiration Date to the current time plus 1 hour

5.9.2 Starting the Security Server in Locksmith Mode

You can start only the master security server in locksmith mode. You start the security server locally, and from the command line rather than the PC-DCE Service Panel. You must shut down PC-DCE before starting the security server from the command line.

Use the following form of the **secd** command to start a security server in locksmith mode:

secd [**-locksm**[**ith**] *pname* [**-lockpw**]]

where:

-locksm[**ith**] – Starts a security server in locksmith mode.

pname – Specifies the name of the locksmith principal. If no registry account exists for this principal, **secd** creates one.

-lockpw – Prompts for a new locksmith password. This option allows you to specify a new password for the locksmith account when the old one is unknown.

To start the master security server in locksmith mode:

1 From the PC-DCE Service Panel, shut down PC-DCE on the master security server machine.

2 Start the security server in locksmith mode. The following example shows the security server started with the locksmith account that was created for the principal named **houdini**:

```
C:\> secd -locksmith houdini
```

If the locksmith account does not exist, it prompts you to create it. Respond **y** and supply passwords to create the new account:

```
Account for houdini doesn't exist. Should this be created ? [y/n]? <y>y
Enter password for locksmith account:
Reenter password to verify:
```

NOTE: If the account exists but you have lost its password, use the **-lockpw** option to cause **secd** to prompt you for a new locksmith password and replace the existing password with the one you enter.

The security server (**secd**) remains running in foreground mode. Leave this command prompt window open.

3. Open another command prompt window and log into DCE as the locksmith:

   ```
   C:\> dce_login houdini bagoftricks
   ```

4. Repair the registry damage.
5. Stop the security server:

   ```
   C:\> sec_admin
   Default replica: /.../longwood
   Default cell:    /.../longwood
   sec_admin> stop
   sec_admin> exit
   bye.
   ```

6. Destroy the locksmith credentials:

   ```
   C:\> kdestroy
   ```

7. Close the command prompt windows.
8. Restart PC-DCE from the PC-DCE Service Panel.

5.10 Handling Security Replica IP Address Changes

If you change the IP address of a master or slave security replica, you must update the **pe_site** files on all security server systems.

The **pe_site** file is located in *install_directory***opt\dcelocal\etc\security** file on that host before restarting PC-DCE. This file lists the IP address of each security replica in the cell. There are two entries for each server.

After you change the IP address of the machine, open the **pe_site** file for each replica in the cell and update the two entries for the changed replica.

For example, if your cell contains two servers named **banks** and **darwin**, and darwin is a slave, the **pe_site** file on **darwin** might look like this:

```
/.../longwood8 00532910-5db6-153f-bf58-00802964ff95@ncadg_ip_udp:
194.90.110.141 subsys/dce/sec/darwin.myco.com
/.../longwood8 00532910-5db6-153f-bf58-00802964ff95@ncacn_ip_tcp:
194.90.110.141 subsys/dce/sec/darwin.myco.com
/.../longwood8 00532910-5db6-153f-bf58-00802964ff95@ncadg_ip_udp:
194.90.110.53 subsys/dce/sec/master
/.../longwood8 00532910-5db6-153f-bf58-00802964ff95@ncacn_ip_tcp:
194.90.110.53 subsys/dce/sec/master
```

Suppose you change darwin’s IP address to 194.90.110.80. You would change the **pe_site** file as follows:

```
/.../longwood8 00532910-5db6-153f-bf58-00802964ff95@ncadg_ip_udp:
194.90.110.80 subsys/dce/sec/darwin.myco.com
/.../longwood8 00532910-5db6-153f-bf58-00802964ff95@ncacn_ip_tcp:
194.90.110.80 subsys/dce/sec/darwin.myco.com
/.../longwood8 00532910-5db6-153f-bf58-00802964ff95@ncadg_ip_udp:
194.90.110.53 subsys/dce/sec/master
/.../longwood8 00532910-5db6-153f-bf58-00802964ff95@ncacn_ip_tcp:
194.90.110.53 subsys/dce/sec/master
```

5.11 Using the Name Service Interface Gateway

The Name Service Interface Gateway (**nsid**), allows remote systems that only have RPC services to use the DCE CDS name service. The **nsid** runs on one or more DCE systems in the cell and acts on behalf of the remote system to execute the RPC name service API calls. Through a hidden level of indirection, the **nsid** allows the PC to appear as if it is directly involved in the broader cell namespace.

5.11.1 Configuring the nsid

To configure **nsid**, your system must also be a CDS client or a CDS server. For procedural information about configuring, starting, or stopping **nsid**, see the online help system that accompanies the PC-DCE Configuration Panel.

During configuration, the appropriate **nsid** settings are defined in the Windows registry. Once configured, the **nsid** is added to your DCE configuration and is started along with all other DCE components.

5.11.2 Security Considerations

RPC communication between the client system (using the services of the **nsid**) and the system running the **nsid** uses unauthenticated Microsoft RPC. The **nsid** runs under the fixed principal, pc-user. Communication between the system running the **nsid** and the DCE Cell Directory Service is authenticated under this principal.

In order for the **nsid** to access entries in the DCE namespace on behalf of the client system, you must modify the access control lists (ACLs) on the namespace entries to authorize access by the **nsid** principal. However, if the namespace entry .:/subsys/DEC/pc is used by the client system, you do not need to modify the ACLs.

The ACLs are preset with authorized access for the **nsid** principal pc-user. For example, a MSRPC server can export an interface named "foo" with the cds entry name ".:/subsys/DEC/pc/foo", without modifying the ACLs. A MSRPC client can then import a binding to that interface using the same cds entry name.

For Windows NT and Windows 2000, **nsid** does use DCE security. A security Principle and Account must be created on the CDS Server with rgy_edit (or rgyedit for a Windows NT or Windows 2000 Server):

```
hosts/your_PC_name/pc-user
```

Then, on the client side, you must create a security key with rgyedit:

```
ktadd -p hosts/<your PC name/pc-user -pw <principle password>
```

5.11.3 The Microsoft Locator and the nsid

The Locator is Microsoft's simple, flat-namespace directory service. The Locator exports the Microsoft version of the RPC name service interface (NSI) and makes an association between entry-name strings and string bindings.

The Locator exports the identical interface as the **nsid**. The caller of the Microsoft NSI makes a remote procedure call to either the Locator or the **nsid** based entirely on the string-binding components defined at the time in the Registry.

5.11.4 The Microsoft Registry and the nsid

The Registry defines the name server that will be queried when any of the rpc_ns_* procedures is called. The name server can be either the Microsoft Locator or the **nsid**; after installation of Windows NT or Windows 2000, the default setting is the Locator. The following section describes how to modify your Regisry to employ a remote **nsid**.

5.11.5 Modifying the Windows NT Registry for Using the nsid

On Windows NT and Windows 2000, the Registry is an integral component of the operating system. There are two methods for modifying the Registry to define settings for **nsid**.

- The Network applet in the Control Panel
- The Registry Editor

5.11.5.1 To Modify the Registry Through the Windows Control Panel:

The RPC Name Service Provider is usually changed through the Network applet in the Control Panel.

To perform this task, do the following:

1. Start the Network applet from the Windows Control Panel.
2. Under the Services tab, select RPC Configuration then click Properties.
3. Under Name Service Provider, select DCE Cell Directory Service.
4. Supply the IP address (or the DECnet address) in the Network Address box.

5 Click OK to change the configuration, then click close to exit the applet.

The machine must be rebooted for the changes to take effect.

5.11.5.2 To Modify the Registry by Using the Registry Editor:

Another way to modify the Registry is to use the registry editor (regedt32.exe). You must use this method to add the default name_service entry.

To set up the Registry on Windows NT or Windows 2000 to use the nsid, do the following:

1 From a command window, enter regedt32 to display the Registry Editor window.

2 Place the focus in the HKEY_LOCAL_MACHINE subwindow by clicking it.

3 Double-click successively on SOFTWARE, Microsoft, Rpc, and NameService. The right half of the HKEY_LOCAL_MACHINE subwindow will now list the parameters for the RPC Name Service Provider.

4 Change the settings as follows:

NOTE: You change the values by double-clicking them; a window appears that allows you to change the value.

Setting	New Value
DefaultSyntax	Either 0 or 3
Endpoint	No value
NetworkAddress	Network address of the system where the **nsid** will be running
Protocol	One of the protocol sequences that the Windows NT system will use to communicate with the **nsid** (ncacn_ip_tcp or ncadg_ip_udp
ServerNetworkAddress	Network address of the system where the **nsid** will be running

After the task is completed, the right half of the HKEY_LOCAL_MACHINE subwindow displays parameter information similar to the following:

```
DefaultSyntax:REG_SZ:3
Endpoint:REG_SZ:
NetworkAddress:REG_SZ:16.64.0.79
Protocol:REG_SZ:ncacn_ip_tcp
ServerNetworkAddress:REG_SZ:16.64.0.79
```

To finish setting up the Registry, pull down the Registry item on the menu bar and choose Exit.

CHAPTER 6

Advanced Client Configuration

Refer to the PC-DCE Configuration Panel help and the DCEsetup help files for general information about configuring your cell. This chapter provides advanced client configuration information and contains the following sections:

6.1 Using timesync.exe to Synchronize System Time

PC-DCE includes a **timesync.exe** process that automatically synchronizes a Windows system’s clock with a DTS Server running on the Security Server every time Windows is started. The system will always remain within the defined five minute window that DCE security mandates. This process reduces the need to run **dtsd** on Windows clients.

NOTE: The Distributed Time Service (**dtsd**) requires at least three DTS servers per cell to ensure accuracy. For **timesync** to work on client systems, at least one DTS server must be running on the same machine as the master security server.

Each time a system is started, **timesync** synchronizes the system clock; however, if the system is not restarted for a long period of time and drifts outside of the five minute window, **timesync** must be run manually to resynchronize. Otherwise, the user will be unable to log into PC-DCE from that system.

NOTE: On Windows 2000 or Windows NT clients, users must have the Change System Time permission to run **timesync** manually. Members of the Administrators group have this permission by default, but you may need to grant this right to users who are not members of the Administration group.

To grant rights to change the system time, create a group for the users who need this right. Then:

In Windows NT or Windows NT Terminal Server:

1. Access the Windows User Manager (**Start > Programs > Administrative Tools > User Manager**).
2. Select **Policies > User Rights**.
3. In the User Rights Policy dialog, add the group of users and assign the Change the System Time right to the group.

In Windows 2000 or Windows 2000 Terminal Server:

1. From the Windows Control Panel, access Local Security Policy (**Start -> Control Panel -> Administrative Tools -> Local Security Policy**).
2. Double-click **Local Security Policies**.
3. Click **User Rights Assignment**.
4. Click **Change the system time**.
5. Click **Add**.
6. Select relevant groups or users.

6.2 Controlling CDS Cache Operations

6.2.1 CDS Cache Overview

The CDS cache is a collection of information about servers, clearinghouses, and other CDS resources that a CDS clerk establishes on the local system for its reference. When the CDS clerk receives a CDS server's response to a query, it stores the response in its cache. The next time the clerk needs this information, the clerk retrieves it from the cache rather than issuing a network request to a CDS server.

6.2.1.1 Cache Implementation

The CDS Cache is maintained in two areas: common (global) cache and per-user cache.

- The common cache contains information available to everyone (for example cell name, directory entries, and clearinghouses). This file, called **cds_cache.000000000X** (X represents a number that increments), is located in **PCDCE32\opt\dcelocal\var\adm\directory\cds.**

  The common CDS cache information is protected by DCE ACLs.

- The per-user cache contains user-specific information (softlinks, groups, etc.). This file, called *cds_server_name***_cache.00000000X** (X represents a number that increments), is located in **PCDCE32\opt\dcelocal\var\adm\directory\cds\***machine name*.

  Per-user information is protected against unauthorized access by DCE ACLs and, if the machine is using the NTFS file system, by NTFS file security.

In PC-DCE , the CDS Clerk is implemented as a DLL whereas the CDS Advertiser is a separate process (**cdsadv**). The cache and advertiser interconnect using an Windows-specific interprocess communications protocol.

The cache is periodically dumped to disk in the set of files stored in *install_directory***opt\dcelocal\var\adm\directory\cds\*.***.

6.2.1.2 Lifetime of Cached Information

Information remains stored in the cache until either of the following occurs:

- The lifetime (roughly 8-10 hours) of the cached entry expires.
- The user establishes new credentials (probably through a new **dce_login**). This updates both the common and per-user cache files.

For example, if a user accesses a CDS server first thing in the morning to locate the services, subsequent lookups during the remainder of the day take advantage of the cache. The next morning, this whole process takes place again because the cached data has become stale or the user has logged into DCE again.

None of the CDS cache information stays fresh indefinitely.

6.2.1.3 CDS Cache Size

The CDS Cache Size is tunable. To set the CDS Cache Size, use the CDS_CACHE_SIZE system environment variable. If this variable does not already exist, you must create it.

To create the CDS_CACHE_SIZE environment variable:

1. Log on to Windows using an account with Administrator privileges.
2. Open the Control Panel, double-click the System icon, and click the Environment tab.
3. In the Environment dialog, click anywhere in the System Variables list.
4. In the Variable text box, type CDS_CACHE_SIZE.
5. In the Value text box, type the cache size (in bytes). Specify the size of the cache in bytes between 131072 (128KB) and 16777216 (16MB).
6. Click Set and then click OK.

The increased cache will be available the next time you start your computer.

6.2.1.4 How a Client Selects a Clearinghouse

When a client needs to read data from CDS, it contacts a clearinghouse. Because a cell can contain many clearinghouses, the client needs a mechanism to help it choose a clearinghouse based on clearinghouse performance and availability. For example, for performance reasons the client ideally uses a clearinghouse connected to the local LAN, but may need to connect to another clearinghouse when the local clearinghouse is down.

The client selects a clearinghouse from a list of clearinghouses stored in the client's CDS cache. The client keeps the list sorted in an order that keeps clearinghouses that are both local (on the same LAN) and available at the top of the list. The client attempts to contact clearinghouses in the order that they appear in the list.

6.2.1.5 How a Client Builds the Clearinghouse List

The CDS cache contains a section with entries for each clearinghouse that it knows about. It learns about clearinghouses in the following ways:

- Configuration. A client's cache always contains the clearinghouse of the preferred CDS server, which is the CDS Server Host Name that you enter when running the configuration program.
- Advertisements. Full clients learn about a new clearinghouse when the CDS server that maintains that clearinghouse issues broadcast advertisements. This works only for full clients, and only for new servers on the same LAN.
- Root Directory. A client generally reads the CDS root directory when it initializes its cache, and thereafter when this cached information expires. This is how full clients learn about off-LAN clearinghouses, and how lightweight clients learn about all new clearinghouses.

6.2.1.6 How a Client Sorts the Clearinghouse List

Each clearinghouse entry in the cache is marked with two flags: *OK* and *OnLAN*. The OK flag indicates whether or not the clearinghouse was last known to be responding to requests (available), and the OnLAN flag indicates whether or not the clearinghouse is located on the local LAN.

The client sorts the list in the following order:

1. OK and OnLAN
2. OK and not OnLAN
3. Not OK

The client keeps the OK and OnLAN flags updated using the following methods:

- The CDS advertiser (**cdsadv**) sets the OnLAN flag for a clearinghouse if it receives a broadcast from that clearinghouse. This makes sense because broadcasts do not go beyond the local LAN.
- Also, at configuration time, the CDS clerk sets the OnLAN flag for the clearinghouse associated with the preferred CDS server. If a user is running the lightweight client (no **cdsadv** or **dced**), then this becomes the only server that is flagged as OnLAN.

Note that, at configuration time, the CDS clerk sets the OnLAN flag for the preferred CDS server, even if the preferred CDS server is outside of the LAN. **dce_update** pings servers that are marked Not OK. If the preferred server (outside the LAN) goes down, **dce_update** continues to ping it, allowing the server to be sorted to the top of the cache again when the server comes back up.

When a clearinghouse entry is added to the cache, the entry is initially flagged as OK. The CDS clerk flags the entry as Not OK if it tries to reach the clearinghouse but there is no response. The entry gets set to OK again in one of the following ways:

- The CDS advertiser receives a broadcast from the clearinghouse. This only works for full clients, and only if the clearinghouse is located on the local LAN.
- The CDS clerk has tried all of the clearinghouses flagged as OK, starts trying clearinghouses in the Not OK section of the list, and receives a response from the clearinghouse.
- The **dce_update** process solicits this clearinghouse and it responds. This works for full and lightweight clients, and works whether or not the clearinghouse is located on the local LAN. **dce_update** solicits clearinghouses on a periodic basis controlled by the **CDSUpdateInterval** registry key (refer to *Section A.1 on page 117*).

6.2.2 Tuning the Update Rate of the Cached Clearinghouse List

You can tune the rate at which **dce_update** solicits clearinghouses by editing values in the registry key \HKEY_LOCAL_MACHINE\SOFTWARE\Entegrity\DCE\Configuration:

- **CDSUpdateInterval**

  REG_DWORD

  dce_update periodically solicits the clearinghouses listed in the cache that are marked Not OK (see *Section 6.2.1.6 on page 94*). CDSUpdateInterval sets the interval between updates, in seconds. The default interval is one hour (3600). Any value greater than zero is acceptable. For example, if an interval of 3 hours is desired, set this tunable to 10800.

  You must restart PC-DCE after setting this value. Notice that if NoCDSUpdateThread is set, this value is ignored.

- **NoCDSUpdateThread**

  REG_DWORD

  Disables the clearinghouse solicit function in **dce_update**. Only 0 and 1 are acceptable as values. If set to a value of 1, **dce_update** will not update the cache; if set to a value of 0, **dce_update** solicits clearinghouses as specified by CDSUpdateInterval. The default value is 0.

  You must restart PC-DCE after setting this value.

6.2.3 Using CDS Preferencing to Control Sorting of the Clearinghouse List

The CDS preferencing feature lets you assign integer ranks to clearinghouses. The ranks affect the sorting of the cached clearinghouse list.

6.2.3.1 How CDS Preferencing Works

You assign ranks to clearinghouses in a *preference file*, which PC-DCE reads at startup. Ranks with lower values are preferred.

If the file does not contain an entry for a particular clearinghouse, PC-DCE calculates a rank. The calculation is based on IP address:

- Clearinghouses with same IP address as local host: rank = 5000
- Clearinghouses on same IP subnet as local host: rank = 20000
- Clearinghouses on same IP network as local host: rank = 30000
- All other clearinghouses: rank = 40000

If the file *does* contain an entry for a particular clearinghouse, this is considered an *override*. Clearinghouses with override ranks are sorted a little differently from clearinghouses with calculated ranks, as described below.

Section 6.2.1.6 stated that the client sorts the cached clearinghouse list in the following order:

1. OK and OnLAN
2. OK and Not OnLAN
3. Not OK

Ranks affect this sorting as follows:

1. OK and OnLAN, as well as OK and Not OnLAN clearinghouses with override ranks. This section of the list is not sorted any further.
2. OK and Not OnLAN. This section of the list is sorted by rank.
3. Not OK. This section of the list is not sorted any further.

6.2.3.2 Creating the Preference File

The preference file is a text file named **cds_serv_pref** located in *install_directory***\opt\dcelocal\etc**. The file contains a series of one-line entries, where each entry consists of a clearinghouse name and the rank for that clearinghouse.

Specify the clearinghouse name (*name_ch*) using one of the following formats:

```
/.../cellname/name_ch
```

```
/name_ch
```

```
name_ch
```

```
/.:/name_ch
```

If the clearinghouse's cellname is not specified, the local cell is assumed.

The rank is a 16-bit unsigned integer (range 0x0000 [0] - 0xffff [65535]). A lower number indicates a stronger preference. A rank of 65535 specifies that the clearinghouse is never to be contacted.

Specify the rank in decimal, octal (with leading "0") or hex (with leading "0x").

Blank lines are ignored. You can include comments after the “#” character.

Example file:

```
# This is an example preference file
/.:/foo_ch               50  # most preferred clearinghouse
/.:/bar_ch              100
/.../mycellname/baz_ch  100
```

When you create or edit the preference file, you must:

1 Stop PC-DCE.

2 Delete the CDS cache file (*install_directory***opt\dcelocal\var\adm\directory\cds\*.***).

3 Restart PC-DCE.

6.2.3.3 Viewing Ranks

You can view the rank of a cached clearinghouse on a full client using the following commands:

```
dcecp -c cdscache show -clearinghouse /.:/name_ch
```

```
cdscp show cached clearinghouse /.:/name_ch
```

6.2.3.4 CDS Preferencing Example

Suppose a client’s preference file is as follows:

```
/.:/a_ch              100  # most preferred clearinghouse
/.:/b_ch              200  # preferred local backup
/.:/c_ch              500  # preferred off-LAN backup
```

Table 6-1 demonstrates cache sorting based on clearinghouse flags and ranks:

Table 6-1: Demonstration of Cache Sorting

Sort Order	Clearinghouse	Flags	Rank
1	a_ch	OK, OnLAN	100 (override)
2	c_ch	OK, Not OnLAN	500 (override)
3	d_ch	OK, OnLAN	20000
4	e_ch	OK, Not OnLAN	30000
5	f_ch	OK, Not OnLAN	40000
6	g_ch	OK, Not OnLAN	40000

Table 6-1: Demonstration of Cache Sorting

Sort Order	Clearinghouse	Flags	Rank
7	h_ch	Not OK, Not OnLAN	40000
8	i_ch	Not OK, Not OnLAN	40000
9	j_ch	Not OK, Not OnLAN	40000
10	b_ch	Not OK, OnLAN	200 (override)
11	l_ch	Not OK, Not OnLAN	40000

The first section of the sorted cache includes all OK and OnLAN clearinghouses, as well as OK and Not OnLAN clearinghouses with override ranks. In the example, this section contains three clearinghouses: a_ch, c_ch, and d_ch. Clearinghouse a_ch is the client's most preferred clearinghouse according to its override rank of 100. Clearinghouse c_ch is sorted next even though it is off-LAN because of its override rank of 500. The final entry in the first section is clearinghouse d_ch, which has an automatically assigned rank.

The second section of the sorted cache includes OK and Not OnLAN clearinghouses sorted by rank. In this example, all clearinghouses in this section have automatically assigned ranks. Notice that clearinghouse e_ch is sorted to the top of this section because it is on the client's subnet and therefore has a rank of 30000.

The third section of the sorted cache includes all Not OK clearinghouses and is unsorted. Notice that clearinghouse b_ch appears in this section despite its low rank. When b_ch comes back online and the client detects this, the client will move b_ch to the first section of the cache.

6.2.4 Refreshing Cached Application Server Data

If you are concerned your client is using stale CDS information, you can update the cache as follows:

- Stop DCE services, delete the cache files, and restart DCE services. Global cache and per-user cache are stored in separate files (see *Section 6.2.1.1*) so you can refresh them individually or delete both files at once.

  This is a brute-force method that effectively resets the cache. However, once you reset the cache using this method, the new cache will initially contain only a single clearinghouse reference (the preferred CDS server, which is the CDS Server Host Name you entered for this client at configuration time). If this server is down or fails before the cache has been repopulated, the client will not be able to fail over to another clearinghouse. Repopulation of the cache occurs when the cache re-initializes (after 8 hours or when the user logs in again), or gradually as on-LAN servers issue broadcasts.

- Programmatically reduce the expiration age to a small value using **rpc_ns_mgmt_set_exp_age()** and then restore the default value.

  This can be useful in a dynamic environment like a development test lab where cache entries may become stale before they are refreshed. For example, if a server uses Object UUID's and is stopped and restarted while its clients are bound, that new instance of the server is not known to the local client cache. The client cache is stale and the client will not be able to find the server.

  You can use **rpc_ns_mgmt_set_exp_age()** to refresh the cache every time your client application is started or after trapping the case where **rpc_mgmt_is_server_listening()** fails. In each case, you can use the call to immediately refresh your cache and then if necessary return to the default cache expiration (usually several hours).

  For details on using this call, refer to the *OSF DCE Application Development Guide*.

You can use the CDS Preferencing feature to control which CDS clearinghouses that the client runtime queries.

6.3 Controlling Client Selection of Security Servers

Normally, the runtime looks up bindings for a security server by using CDS. However, if CDS is unavailable, the client runtime selects a security server based on the contents of the **pe_site** file, which is a list of security servers and associated bindings. The runtime starts by trying to contact the server listed at the beginning of the **pe_site** file. If that server fails to respond, the runtime tries the next server listed in the file. **dce_update** dynamically updates **pe_site** to keep the list sorted based on server availability. You can control the rate of update as described in *Section 6.3.2*.

In early releases of DCE, **pe_site** was static and you could control runtime selection of security servers by editing this file. Now, by default **pe_site** is updated dynamically. You can disable dynamic update using the registry key **NoSECUpdateThread** as described in *Section 6.3.2*. However, PC-DCE also offers an alternative method of specifying a preferred security server: setting up a special RPC profile, as described in *Section 6.3.3 on page 100*. This method allows you to retain the advantage of dynamic update.

6.3.1 Forcing the Runtime to Use pe_site

You can force the runtime to use **pe_site** exclusively, rather than CDS, by setting the environment variable BIND_PE_SITE to 1.

6.3.2 Controlling the Rate of pe_site File Updates

dce_update periodically pings all known security servers and moves servers that do not respond to the bottom of the list contained in the **pe_site** file. You can tune the rate at which **dce_update** solicits security servers by editing values in the registry key:

```
/HKEY_LOCAL_MACHINE/SOFTWARE/Entegrity/DCE/Configuration
```

The keys to edit are as follows. By default, these keys do not exist, so you need to create them.

- **SECUpdateInterval**

  REG_DWORD

  SECUpdateInterval sets the interval between **pe_site** updates, in seconds. The default interval is one hour (3600). Any value greater than zero is acceptable. For example, if an interval of 3 hours is desired, set this value to 10800.

  You must restart PC-DCE after setting this value. Notice that if NoSECUpdateThread is set, this value is ignored.

- **NoSECUpdateThread**

  REG_DWORD

  Enables or disables the **dce_update** function that dynamically updates the **pe_site** file. Only 0 and 1 are acceptable as values. If set to a value of 1, **dce_update** will not update the **pe_site** file; if set to a value of 0, **dce_update** updates **pe_site** according to SECUpdateInterval.

  You must restart PC-DCE after setting this value.

6.3.3 Using an RPC Profile to Specify a Preferred Security Server

Because the **pe_site** file is updated dynamically, you cannot use it to specify a preferred security server unless you disable the dynamic update function. An alternative method of specifying that a client use a particular security server is to set up an alternate RPC profile in the CDS that points to this server, and then set up the client registry key SEC_DEFAULT_ENRY to point to this profile.

Notice that you can create multiple alternate profiles and use SEC_DEFAULT_ENTRY to point to the preferred profile for each client system.

To specify a preferred security server:

1 You need a cut-and-paste source for the interface identifiers for each of the security services. If you are reading this document online, you can cut and paste them from the listing of the **rpcprofile show** command example below. (You can do this because the interface identifiers for the security services are static, and are the same in this document as they will be if you obtain them from another source.) Otherwise, you must run the **rpcprofile show** command yourself to view the current cell profile, and then copy the output of this command into a convenient spot from which to cut and paste, such as an open text file.

```
C:\> dce_login cell_admin -dce-
C:\> dcecp
dcecp> rpcprofile show /.:/cell-profile
{{d46113d0-a848-11cb-b863-08001e046aa5 2.0} /.../longwood/sec 0 rs_bind}
{{0d7c1e50-113a-11ca-b71f-08001e01dc6c 1.0} /.../longwood/sec-v1 0
secidmap}
{{8f73de50-768c-11ca-bffc-08001e039431 1.0} /.../longwood/sec 0 krb5rpc}
{{b1e338f8-9533-11c9-a34a-08001e019c1e 1.1} /.../longwood/sec 0 rpriv}
{{b1e338f8-9533-11c9-a34a-08001e019c1e 1.0} /.../longwood/sec 0 rpriv}
{{6f264242-b9f8-11c9-ad31-08002b0dc035 1.0} /.../longwood/lan-profile 0
LAN}
{{4d37f2dd-ed43-0000-02c0-37cf2e000001 4.0} /.../longwood/fs 0 fs}
{{eb814e2a-0099-11ca-8678-02608c2ea96e 4.0} /.../longwood/subsys/dce/dfs/
bak 0 bak}
{{1edd9c80-eed7-1299-a70f-0000c0e26f5f 1.1} /.../longwood/sec 0
Entegrity/rdlg_tok}
```

2 To obtain the exact server name of the preferred server, list the names of all security servers by listing the members of the RPC group **/.:/sec**:

```
dcecp> rpcgroup list /.:/sec
/.../longwood/subsys/dce/sec/master
/.../longwood/subsys/dce/sec/darwin.explorers.com
```

3 Create the profile:

```
dcecp> rpcprofile create /.:/my_profile
```

4 Add entries to this profile that specify your preferred server for all of the required security interfaces.

In this example we select **darwin.explorers.com** as the preferred server. The required interfaces are the first five that appear in the **cell-profile** listing (*Step 1*). To add an entry, use the following dcecp command:

rpcprofile add *profile_name*
  -m *server_name*
  -interface *interface_id*
  -priority *n*

NOTE: The priority setting is useful only if your profile includes multiple entries for a specific interface. For details, refer to the *OSF DCE Command Reference*.

For example, to add entries for the required services:

rs_bind:

```
dcecp> rpcprofile add /.:/my_profile
  -m /.:/subsys/dce/sec/darwin.explorers.com
  -interface {d46113d0-a848-11cb-b863-08001e046aa5 2.0}
  -priority 0
```

secidmap:

```
dcecp> rpcprofile add /.:/my_profile
  -m /.:/subsys/dce/sec/darwin.explorers.com
  -interface {0d7c1e50-113a-11ca-b71f-08001e01dc6c 1.0}
  -priority 0
```

krb5rpc:

```
dcecp> rpcprofile add /.:/my_profile
  -m /.:/subsys/dce/sec/darwin.explorers.com
  -interface {8f73de50-768c-11ca-bffc-08001e039431 1.0}
  -priority 0
```

rpriv (v1.1):

```
dcecp> rpcprofile add /.:/my_profile
  -m /.:/subsys/dce/sec/darwin.explorers.com
  -interface {b1e338f8-9533-11c9-a34a-08001e019c1e 1.1}
  -priority 0
```

rpriv (v1.0):

```
dcecp> rpcprofile add /.:/my_profile
  -m /.:/subsys/dce/sec/darwin.explorers.com
  -interface {b1e338f8-9533-11c9-a34a-08001e019c1e 1.0}
  -priority 0
```

5 Add a default entry specifying the master security server. This is necessary because if the client is unable to contact a service specified in the alternate profile, it can fall back to the default entry.

```
dcecp> rpcprofile add /.:/my_profile
  -m /.:/subsys/dce/sec/darwin.explorers.com
  -default
```

6 On the client, add the SEC_DEFAULT_ENTRY registry key as a subkey to the key:

```
/HKEY_LOCAL_MACHINE/SOFTWARE/Entegrity/DCE/Configuration
```

Set the value to the name of your alternate profile. For example:

```
SEC_DEFAULT_ENTRY:REG_SZ:/.:/my-profile
```

CHAPTER 7

Multicell Environments

7

You may want to create an environment in which DCE principals can interact across different cells. For example, an administrator may want to perform DCE administration for several cells from one system. Or, one company may want to grant access to an in-house database to a subset of people at a different company.

This chapter describes how to configure intercell communication in a multicell environment and contains the following sections:

7.1 Understanding Trust Relationships

Trust relationships between cells grant access to principals from one cell to objects in another cell and vice versa. To permit principals in one cell to have authenticated access to objects in another cell, you must establish a trust relationship between the two cells.

Two kinds of trust relationships allow principals in a foreign cell to engage in authenticated access to objects in a local cell. These relationships are:

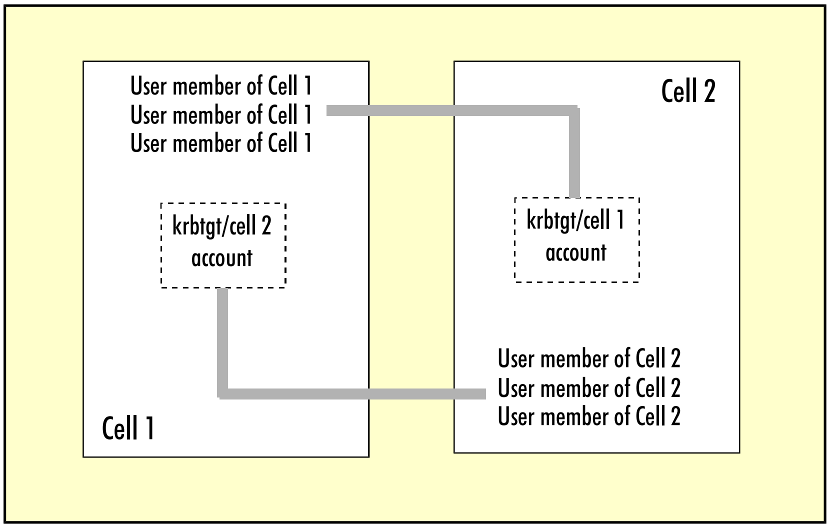
- *Direct trust relationships*
- *Transitive trust relationships* (not supported by OSF DCE Version 1.2.1)

7.1.1 Direct Trust Relationships

A direct trust relationship involves only two cells. The two cells' authentication services share authentication keys and trust each other to authenticate principals from their respective cells. Each cell considers users from the other cell to be authenticated if the user is marked as authenticated by the other cell's authentication service.

Figure 7-1 illustrates the concept of direct trust between two cells. Members of Cell 1 have access to services in Cell 2, on the basis of their membership in the trusted Cell 1, through the account *krbtgt/cell 1*. Likewise, members of Cell 2 have access to services in Cell 1 through the account *krbtgt/cell 2*. The trust relationship also allows users in the foreign cell to log into accounts in the local cell and vice versa.

Figure 7-1: Direct Trust Relationship



7.1.1.1 User Access to the Foreign Cell

From the user’s perspective, access to objects and services in the other cell is transparent and automatic. Once users log into their own cell, they also have access to the foreign cell. No additional login to the cross-cell authentication (krbtgt) account is required.

7.1.1.2 Controlling Authorization for Objects in the Cell

Once trust is established, you can control individual foreign principals’ access to specific objects by editing the ACL entries of local resources. To edit object ACLs, you can use the graphical interface provided by the Visual DCE ACL Editor. See the Visual DCE ACL Editor online help file for additional information. DCE ACLs support three levels of trust for foreign principals:

- **foreign_cell** — Grants or denies access to all users from the foreign cell.
- **foreign_group** — Grants or denies access to foreign users based on group membership.
- **foreign_user** — Grants or denies access to a particular user from the foreign cell.

If you do not specifically grant a foreign cell access to a resource, foreign principals are granted the rights specified by the **any_other** entry (if it exists) in the ACL. You should check the **any_other** rights on any sensitive resources to ensure the resources are sufficiently protected.

7.2 Pros and Cons of Intercell Trust Relationships

Before configuring intercell trust, review the benefits and security implications described in this section.

7.2.1 Simplified Administration

Intercell trust relationships can simplify administration of users by allowing a trusted cell to manage its own security information. For example, Company A allows a cell from Company B to access Company A’s services, and Company B manages the membership of the cell that has privileges to Company A’s services.

Because the cell consists of employees of Company B, Company B is best able to manage cell membership and security (user id and password management, organizational and personnel changes).

7.2.2 Simplicity for the User

For end users, intercell trust provides transparent access between their own cell and the cell to which they have access via the trust relationship. For example, Cell B users can access services from Cell A on the basis of their membership in the trusted Cell B. They are not required to enter a login ID and password for Cell A.

7.2.3 Security Implications

In an intercell trust relationship, one organization is trusting another to manage security information. By administering DCE ACLs, Cell A is able to set access privileges for its resources based on foreign cell membership, foreign cell group membership, or for individual foreign principals (see *Section 7.1.1.2 on page 104*).

However, Cell A is still relying on Cell B to properly authenticate the end user. Cell B clients present privileges to the Cell A server that were generated by the security server in Cell B. These privileges include the client’s DCE identity and the groups to which the client belongs; no further challenges related to the principal’s idenetity are issued by the Cell A server. Thus, setting up an intercell environment makes sense only if Cell A trusts Cell B to properly administer cell membership and security.

7.3 Establishing Intercell Communications

For intercell communication to be successful:

- Both cells must use the same DNS server.
- Each cell's name should be the same as the machine name of its Master Security Server. For example, if the Master Security Server hostname is robin.acme.com, the cell should be called robin.acme.com.
- The master systems must be able to contact each other over the network (to test, ping each system from the other system).

Establishing intercell communications requires you to do the following in the order shown:

7.3.1 Establishing Intercell Lookup

To establish intercell lookup, add entries for each of the two cells to the DNS server:

1. Ensure that the names of the two cells between which you will establish communication include the domain name that the DNS server will use. The cell names should look similar to the following:

   ```
   groucho.eng.entegrity.com
   harpo.eng.entegrity.com
   ```

   Here, `eng.entegrity.com` is the domain name.

2. In the local cell, enter **`show cell as dns`** from the **cdscp** prompt.

   Any system in the cell will supply the correct output for **show cell as dns**; you do not have to use the master system.

   The following example shows the output of this command for the cell `harpo.eng.entegrity.com`:

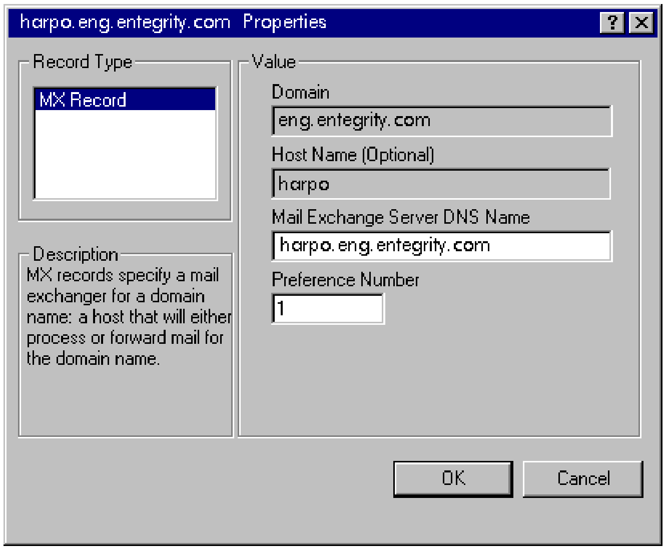
   ```
   cdscp> show cell as dns

          SHOW

          CELL       /.../harpo.eng.entegrity.com

          AT         2000-06-25-12:40:33

          TXT =      1 002cb551-d2d8-b154-00104b9afb74 Master /.../ \
          harpo.eng.entegrity.com/MA1_ch 00282170-d2d8-b9afb74
   ```

3. Refer to the output of the **show cell as dns** command to create two new entries — an MX record and a TXT record — for the local cell in the DNS management program you are using.

NOTE: When you initially configure intercell lookup, include entries for master servers only. After intercell communication is established and running correctly, add DNS entries for replica servers.

- **MX Record** – Gives the entry a preference. Use the cell name, not the machine name, as the hostname of the master server (for example, `harpo.eng.entegrity.com`). Use a preference of 1 for master servers and a lower preference for replica servers.

  The following example shows how an MX record would appear when entered into the Microsoft DNS Manager:



harpo.eng.entegrity.com Properties

Record Type: MX Record

Description
MX records specify a mail exchanger for a domain name: a host that will either process or forward mail for the domain name.

Value
Domain: eng.entegrity.com
Host Name (Optional): harpo
Mail Exchange Server DNS Name: harpo.eng.entegrity.com
Preference Number: 1

- **TXT Record** – Enter the hostname (for example, `harpo`).

  Complete the text field. Use the `TXT =` output from the **`show cell as dns`** command, and add the system hostname at the end of the entry. For example, a text entry might read:

```
1 002cb551-d2d8-b154-00104b9afb74 Master /.../harpo.eng.entegrity.com/
harpo_ch 00282170-d2d8-b9afb74 harpo.eng.entegrity.com
```

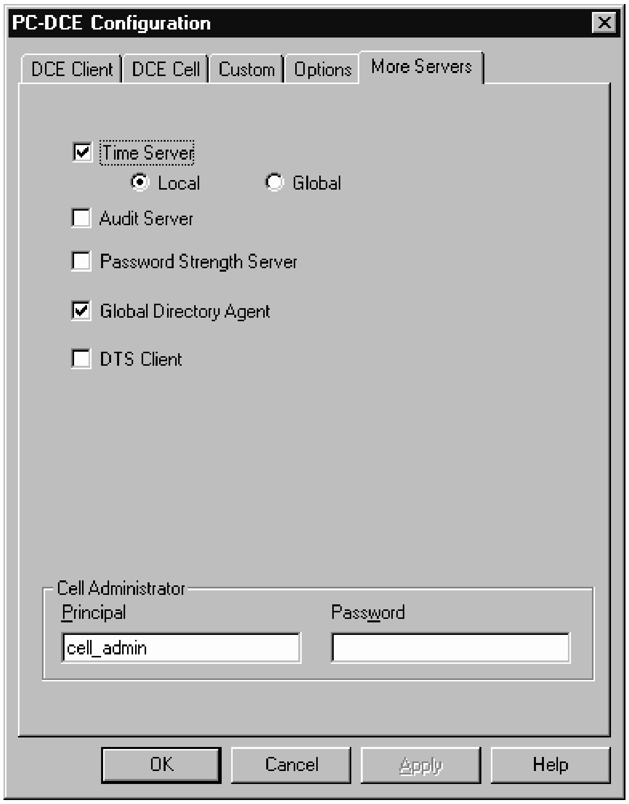
  In this example, `harpo.eng.entegrity.com` is the system hostname that you add to the end of the entry.

NOTE: Ensure that you enter the information exactly as it is displayed in the `TXT =` output of the **`show cell as dns`** command.

4 In the foreign cell, enter **`show cell as dns`** from the **cdscp** prompt. Repeat step 3 to create an MX record and a TXT record for the foreign cell.

5 In each of the two cells, start the GDAD (Global Directory Agent) process.

To do so, access the More Servers tab of the PC-DCE Configuration Panel, enable the Global Directory Agent option. Remember to enter the **cell_admin** password on the More Servers tab:



☑ Time Server

● Local ○ Global

□ Audit Server

□ Password Strength Server

☑ Global Directory Agent

□ DTS Client

Cell Administrator

Principal: cell_admin

Password:

After you click OK, the Configuration Status dialog displays the progress of the configuration.

6 After the GDAD process has been started successfully, click Close to close the Configuration Status dialog.

7 To close the PC-DCE Configuration Panel, click Cancel (this does not cancel the GDAD process).

7.3.2 Establishing Intercell Trust

Establishing a direct intercell trust relationship between a foreign and local cell indicates that you trust the foreign cell’s authentication service to correctly authenticate users in its own cell, and that you consider all users from the foreign cell to be authenticated if they are so marked by the foreign cell’s authentication service.

To establish intercell trust, use **rgy_edit** or **dcecp registry connect** to create an account for the foreign cell in the local cell.

NOTE: If you mistype the parameters for these commands, the account may be created as a local, rather than foreign, account and you will receive error messages when you try to establish the intercell connection.

NOTE: You must create the local and foreign groups and organizations before you run **rgy_edit** or **dcecp registry connect**.

Supply the following parameters with these **rgy_edit** or **dcecp registry connect**:

Foreign cell host name

Foreign account — The foreign account must have the permissions required to create principals and accounts (for example, **cell_admin**). The account accesses the foreign registry to create the account that represents the local cell in the foreign account's registry.

Foreign account's current password

Local group name — Group name to be associated with the account in the local cell (the registry requires all accounts to be associated with a group).

Foreign group name — Group name to be associated with the account in the foreign cell.

Local organization name — Organization name to be associated with the account in the local cell (the registry requires all accounts to be associated with an organization).

Foreign organization name — Organization to be associated with the account in the foreign cell.

Expiration date (optional) — Time and date that both the local and the foreign cell's accounts expire, and the peer-to-peer relationship is ended, prohibiting any further authenticated communications between principals in the two cells. To renew the account, change the date in this field. The default is **none**.

Your password

Example

The following example uses **rgy_edit** to create an account in the local cell for the foreign cell. In this example, the administrator associates the new account with the default group **none** and the default organization **none**. These associations can be changed later.

```
rgy_edit=> cell /.../harpo.eng.entegrity.com
Enter group name of the local account for the foreign cell: none
Enter group name of the foreign account for the local cell: none
Enter org name of the local account for the foreign cell: none
Enter org name of the foreign account for the local cell: none
Enter your password:-dce-
Enter account id to log into foreign cell with: cell_admin
Enter password for foreign account:-dce-
Enter expiration date [yy/mm/dd or "none"]: <none>
Principals and Accounts have been created.
```

The following example shows the same procedure using **dcecp registry_connect**:

```
dcecp> registry connect /.../harpo.eng.entegrity.com -facct cell_admin \
-facctpw -dce- -group none -fgroup none -org none -forg none -mypwd -dce-
dcecp>
```

7.3.2.1 What Happens When You Create an Account for the Foreign Cell

Entering the **rgy_edit** or **registry connect** command does the following:

- Creates two cross-cell authentication accounts and the accounts' principals. One account is created in the local cell's registry to represent the foreign cell, and another account is created in the foreign cell's registry to represent the local cell.

  New account principals are named according to the original cell name and the prefix **krbtgt**; for example, **krbtgt/**<*foreign_cell_name*>.

- Assigns a UUID that is shared by all principals in the foreign cell. Each user in the foreign cell is now identified with the shared cell UUID as well as its own user UUID. The ACL manager uses the cell UUID/user UUID pair to determine access.

  NOTE: In the specific case that you are using DFS with a UFS partition, the system uses only the foreign cell UUID to determine access, not the user UUID. This means that foreign users accessing the UFS partition are seen as the same foreign user, and files on a local system that are owned by a foreign user can be accessed by every other foreign user that is a member of that foreign cell.

The new accounts have the default account attributes described in *Table 3-1 on page 42* except for the following differences:

- **acctvalid** — set to no
- **client** — set to no
- **dupkey** — set to yes
- **postdatedtkt** — set to yes
- **proxiabletkt** — set to yes

These attributes apply to all foreign principals when they access objects in the local cell. Similarly, the attributes of the account created in the foreign cell for the local cell apply to all principals in the local cell when they access objects in the foreign cell.

7.3.3 Verifying Account Creation

To view the new accounts, enter the **dcecp** command **principal catalog**. Each cell now includes two **krbtgt** accounts: its own account (`krbtgt/harpo.eng.entegrity.com`) and the new foreign cell's account (`krbtgt/groucho.eng.entegrity.com`):

```
dcecp> principal catalog
/.../harpo.eng.entegrity.com/nobody
/.../harpo.eng.entegrity.com/root
/.../harpo.eng.entegrity.com/daemon
/.../harpo.eng.entegrity.com/sys
/.../harpo.eng.entegrity.com/bin
/.../harpo.eng.entegrity.com/uucp
/.../harpo.eng.entegrity.com/who
/.../harpo.eng.entegrity.com/mail
/.../harpo.eng.entegrity.com/tcb
/.../harpo.eng.entegrity.com/dce-ptgt
/.../harpo.eng.entegrity.com/dce-rgy
/.../harpo.eng.entegrity.com/cell_admin
/.../harpo.eng.entegrity.com/krbtgt/harpo.eng.entegrity.com
/.../harpo.eng.entegrity.com/hosts/harpo/self
/.../harpo.eng.entegrity.com/hosts/harpo/cds-server
/.../harpo.eng.entegrity.com/hosts/harpo/gda
/.../harpo.eng.entegrity.com/krbtgt/groucho.eng.entegrity.com
```

7.3.4 Modifying the Account Valid Flag

For accounts that represent foreign cells, the account valid flag is set to **no** by default.

After you establish intercell lookup and trust, do the following:

1. Log into the local cell as **cell_admin**.
2. Use **dcecp** to change the **acctvalid** flag of the new account that represents the foreign cell to yes.
3. Log into the foreign cell as **cell_admin**.
4. Use **dcecp** to change the **acctvalid** flag of the new account that represents the local cell to yes.

This establishes that the new accounts are valid. If one or both accounts are invalid, no intercell communications can take place.

For example, log into the cell **groucho** and modify the account valid flag for the **krbtgt/harpo** account as follows:

```
c:\dce_login cell_admin -dce-
Password must be changed!
c:\dcecp
dcecp> account modify krbtgt/harpo.eng.entegrity.com -acctvalid yes
```

To verify that the account valid flag has been changed to yes:

Enter the **dcecp account show** command. This command displays the attributes assigned to the new accounts. Refer to *Section 3.8.4 on page 45* for information about modifying these parameters.

```
dcecp> account show krbtgt/harpo.eng.entegrity.com
{acctvalid yes}
{client no}
{created /.../groucho.eng.entegrity.com/cell_admin\
2000-07-02-10:12:11.000-04:00I-----}
{description {}}
{dupkey yes}
{expdate none}
{forwardabletkt yes}
{goodsince 2000-07-02-10:11:53.000-04:00I-----}
{group none}
{home {}}
{lastchange /.../groucho.eng.entegrity.com/cell_admin\
2000-07-02-10:12:13.000-04:00I-----}
{organization none}
{postdatedtkt yes}
{proxiabletkt yes}
{pwdvalid yes}
{renewabletkt yes}
{server yes}
{shell {}}
{stdtgtauth yes}
```

7.3.5 Adding Entries for Replica Servers to DNS

After you have verified that intercell communication has been successfully established, use the information in *Section 7.3.1 on page 106* to add entries for replica servers to DNS.

CHAPTER 8

Troubleshooting PC-DCE

This chapter describes solutions to runtime problems that may occur with PC-DCE. For troubleshooting information on installation, configuration, or development problems, refer to the appropriate guides.

Table 8-1: Troubleshooting PC-DCE Runtime Problems.

Symptom	Cause	Solutions
Client cannot contact a particular application server. The server is clearly running, and clients running on server machine can contact the server and otherwise work correctly.	The local CDS cache is stale.	Wait until the cache is automatically refreshed. The default refresh interval is about 8 hours.
		Manually update the cache by stopping DCE, deleting the client cache files, and restarting DCE (as described in *Section 6.2.4 on page 98*).
		If the application is under development, you can update the cache programatically by using **rpc_ns_mgmt_set_exp_age()**.
Windows pauses for a long time just before displaying the Windows login dialog. If you press CTRL-ALT-DEL on Windows 98, the Task Manager shows that **dce32init** is not responding. If you press CTRL-ALT-DEL on Windows 2000 or Windows NT, startup simply continues.	DCE cell services not available to client	Make sure the cell is up and running.
		If the cell has been re-configured, re-configure the client.
		If client startup continues to be a problem, it may be because the network is not responding quickly enough. Try reconfiguring the client as follows: in the Options tab in the PC-DCE Configuration Panel, disable **Enable automatic login to DCE** and disable **Start daemons during system boot**
Cannot create security or CDS replica on multi-homed machine.	IP messages not forwarded.	Enable IP forwarding on the replica machine so that IP requests can go from one network card to the next.
When configuring an existing client into a new cell, the configuration program finds and attempts to contact an old security server.	An old PE_SITE file on the client system is pointing to the old security server.	Remove the old **pe_site** file. The default location is **pcdce32\opt\dcelocal\etc\security**. Use the **dcecp** command **rpcentry show /.:/subsys/dce/sec/master** to make sure that this entry is pointing to the current master security server. Configure the client.

Table 8-1: Troubleshooting PC-DCE Runtime Problems. (Continued)

Symptom	Cause	Solutions
Clock skew too great	DCE requires that the client time stay within a five minute window of the time on the security server. Any combination of incorrect time, date, or time zone will produce this error.	Verify that the time and date are accurate. Next, check the time zone setting for the OS (as well as the TZ environment variable should you have one) and confirm that they are correctly honoring or ignoring Daylight SavingsTime.
Decrypt integrity check failed	Cell has been reconfigured.	Reconfigure the client into the cell. This will be required for all clients.
	A ticket has expired.	The DCE security service uses two kinds of tickets: the service ticket and the ticket granting ticket. The service ticket contains a session key which is a temporary secret key used for client-server communication. The ticket granting ticket is used to obtain a service ticket. The tickets have an expiration time, usually two hours. If a ticket has expired but has not yet been refreshed, and a client tries to contact the Security Server, you could see the error. The condition may last only for a matter of milliseconds. Because tickets the client presents to servers will expire and refresh at given intervals, it is possible to see this message occasionally during normal operation. Seeing the message on the server machine does not necessarily indicate there is a problem with the Security Server's ticket, rather it may indicate that a client with an expired ticket has tried to make contact with the Security Server.
	Password on the account that this application server is using has been changed.	Recreate the application server's key table. First, remove the keytab file (for example, **c:\tmp\grade_server_tab**) Then recreate the keytab file (for example): dcecp> **keytab create /.:/hosts/myhost.mycompany.com/config/keytab/grade_server_1 -storage "/tmp/grade_server_tab" -data {grade_server_1 plain 1 -dce-}**
Error with socket (dce/cds)	CDS Advertiser is stopped.	If DCE is started, restart the CDS Advertiser. Otherwise, start DCE.
Malformed representation of principal (dce / krb) (during login or client configuration)	User typed an illegal character as part of principal name.	Try re-entering the name.
	RPC_SUPPORTED_PROTSEQS= ncadg_ip_udp environment variable is set for the server while the client only supports TCP	Remove the **RPC_SUPPORTED_PROTSEQS** environment variable on the server.

Table 8-1: Troubleshooting PC-DCE Runtime Problems. (Continued)

Symptom	Cause	Solutions
Protocol sequence not supported	The underlying network is either unavailable or not working properly.	Confirm that the network is working properly by trying to ping something. If ping fails, investigate the network further.
	The local system is not offering the required protocol sequence.	Make sure that the environment variable RPC_SUPPORTED_PROTSEQS is not set.
Registry Server Not Available: Not Registered in Endpoint Map	The local CDS cache is stale.	Wait until the cache is automatically refreshed. The default refresh interval is about 8 hours.
		Manually update the cache by stopping DCE, deleting the client cache files, and restarting DCE.
		If the application is under development, you can update the cache programatically by using **rpc_ns_mgmt_set_exp_age()**.
Rpc_s_credentials_too_large	The security credentials are too large to fit into an RPC message.	The user belongs to too many groups or is using ERAs. Both increase the size of the security creadentials. Check the group membership for this user. If it is large, try reducing the number of groups. Also check to see if the user has associated ERAs that may be contributing to the size of the credentials.
Rpcss kicker failure (in event log)	Windows NT 4.0 defect.	Install Windows NT 4.0 Service Pack 3.
Runtime Error!	An uhandled exception was raised by the PC-DCE runtime.	Refer to *Developer's Notes*.
Terminal Server	Wrong version of DCE Installed over Microsoft Terminal Server	Install the Terminal Server version of DCE. (See *PC-DCE Installation and Release Notes* section3.4 Installing the Client Runtime Kit)
Unknown Interface	The endpoint mapper does not have a DCE server interface registered which matches the interface the DCE client is requesting. Specifically, the following conditions must be satisfied: ■ Interface UUIDs match. ■ Object UUIDs (if any) match. Major client and server versions match, and the minor version of the client is less than or equal to the minor version of the server.	Restart the server to force it to re-register its bindings with the endpoint mapper.
		Verify that the server is registering the correct interface. Use the **dcecp** command **endpoint show**.

Table 8-1: Troubleshooting PC-DCE Runtime Problems. (Continued)

Symptom	Cause	Solutions
Who are you failed	Specific registry entry not cleaned up during PC-DCE re-configuration	Remove the registry entry.
	Security transaction between client and server failed.	Make sure that both the client and server runtime have the DES option available to them.
		Make sure a server calling **rpc_server_register_auth_info()** has the correct (same) key for both its keytab and matching registry entry.
		Make sure the system times for the client, security server, and application server are within the allowed five minute range.

APPENDIX A

Advanced Configuration Parameters

This appendix describes advanced configuration parameters. It includes the following sections:

A.1 Registry Keys

PC-DCE uses several registry keys that you can modify to fine-tune PC-DCE behavior. These keys are not present in the default configuration; you must create them. In general, if a key is not present, PC-DCE uses the default value.

All keys are subkeys to the key:

```
/HKEY_LOCAL_MACHINE/SOFTWARE/Entegrity/DCE/Configuration
```

SEC_DEFAULT_ENTRY

Discussion	Name of the local profile in the CDS namespace used by the DCE runtime to locate the security server. Refer to *Section 6.3.3 on page 100*.
Default	**/.:/cell-profile**
Example	`SEC_DEFAULT_ENTRY:REG_SZ:/.:/alternate-profile`

SecdWaitTimeout

Discussion	Maximum number of seconds the PC-DCE service waits for the security server to initialize before concluding that a failure has occurred. In testing, a security server with 10,000 accounts in the DCE registry, running on a Pentium 133 with 32 Mbytes of memory, took approximately one minute to initialize and become available.
Default	200 seconds
Example	`SecdWaitTimeout:REG_DWORD:600`

CdsdWaitTimeout

Discussion	Maximum number of seconds the PC-DCE service waits for the CDS server to initialize before concluding that a failure has occurred. In testing, a security server with 2,000 directories and 20,000 objects in the CDS database, running on a Pentium 133 with 32 Mbytes of memory, took approximately five minutes to initialize and become available.
Default	200 seconds
Example	`CdsdWaitTimeout:REG_DWORD:500`

CDSUpdateInterval

Discussion	CDS solicit interval for **dce_update**, in seconds. The shorter the interval, the fresher the cached list of CDS clearinghouses, at the expense of increased network traffic. Notice that if the NoCDSUpdateThread key is set to 1, no updates occur regardless of the value of this key. You must restart PC-DCE after modifying this key. For a more detailed discussion, refer to *Section 6.2 on page 92*.
Default	3600 seconds
Example	`CdsUpdateInterval:REG_DWORD:600`

NoCDSUpdateThread

Discussion	Enables or disables the CDS solicit function in **dce_update**. You must restart PC-DCE after modifying this key. For a more detailed discussion, refer to *Section 6.2 on page 92*.	
Values	0 1	enable disable
Default	Enabled	
Example	`NoCDSUpdateThread:REG_DWORD:1`	

MapNtToDceExceptions

Discussion	Enables or disables NT-to-DCE exception mapping for all DCE daemons and applications running on the system. When mapping is enabled, PC-DCE attempts to handle NT exceptions. Windows NT does not display an error dialog or log the exception in the Event Log. **NOTE:** To enable NT-to-DCE exception mapping for an individual application, use the call **__exc_w32_to_dce_map_set(1)**.
Default	False
Example	`MapNtToDceExceptions:REG_SZ:True`

SECUpdateInterval

Discussion	Security server solicit interval for **dce_update**, in seconds. The shorter the interval, the fresher the contents of the **pe_site** file, at the expense of increased network traffic. Notice that if the NoSECUpdateThread key is set to 1, no updates occur regardless of the value of this key. You must restart PC-DCE after modifying this key. For a discussion, refer to *Section 6.3.2 on page 100*.
Default	3600 seconds
Example	`SECUpdateInterval:REG_DWORD:600`

NoSECUpdateThread

Discussion	Enables or disables the security server solicit function in **dce_update**. You must restart PC-DCE after modifying this key. For a discussion, refer to *Section 6.3.2 on page 100*.	
Values	0 1	enable disable
Default	Enabled	
Example	`NoSECUpdateThread:REG_DWORD:1`	

A.2 Environment Variables

This section provides information about environment variables that can be modified for use with PC-DCE.

A.2.1 Variables for Tuning sec_key_mgmt_manage_key()

PC-DCE provides environment variables you can use to modify the **sec_key_mgmt_manage_key()** API function. This function is used by DCE daemons to manage their respective keys.

The **sec_key_mgmt_manage_key()** function by default checks password information for a principal as noted below:

- If a principal’s password has an expiration date, this API function would wake up 10 minutes (the default) before the password is due to expire.
- If a principal’s password does not have an expiration date, this API function would continue checking every 10 minutes.

You can set environment variables by opening the Control Panel, selecting the System icon and clicking the Environment Tabbed Dialog.

DCE_SEC_KEYMGMT_WAKEUP_INTERVAL

Represents how often (in seconds) that the **sec_key_mgmt_manage_key()** function should check a principal’s password information to verify that its expiration hasn't been changed.

For example, on a DCE client all the daemons run under the machine/host principal **hosts/***hostname***/self**. The password for this machine principal is set so that it will never expire. So, if DCE_SEC_KEYMGMT_WAKEUP_INTERVAL is set to a value of 7200 (2 hours), all the daemons will check with their Security servers about changes in password expiration settings once every 2 hours instead of the default of every 10 minutes.

DCE_SEC_KEYMGMT_GRACE_PERIOD

Represents the grace period (in seconds) during which the **sec_key_mgmt_manage_key()** function should check a principal’s password information before the password is due to expire.

Assume that the password expiration for an application server principal is set to 5 hours.

By setting the DCE_SEC_KEYMGMT_WAKEUP_INTERVAL to 10800 (3 hours) and setting DCE_SEC_KEYMGMT_GRACE_PERIOD to 300 (5 minutes), the **sec_key_mgmt_manage_key()** will check with the security server once every 3 hours and wake up 5 minutes before the key is due to expire.

A.2.2 Variable for Multi-homed Machines

RPC_UNSUPPORTED_NETIFS

Contains a list of TCP/IP addresses that PC-DCE should not export bindings on. This variable may be useful in multi-homed machines. The list should be space delimited; for example:

```
192.93.110.1 205.67.164.5
```

APPENDIX B

Understanding Client Failover

If a client’s preferred CDS or security server is unavailable, clients will automatically locate (*fail over* to) an available replica. This section provides performance expectations for client failover in a standard Windows NT 4.0 environment, and describes how other environmental factors influence the speed and success of client failover.

This section describes:

B.1 Situations that Trigger Failover

If a CDS or security server is unavailable when one of the following actions occurs, the client fails over to a backup server:

- Client logs into DCE
- Client credentials expire and client requests refreshed credentials
- Client makes a request of an application server, and the application server contacts the security server to verify the client’s credentials
- Client performs a directory service lookup

B.1.1 Situations That Do Not Trigger Failover

This section discusses situations in which failover may seem to occur, but in fact is not really occurring. Understanding PC-DCE behavior in these situations will help you develop good failover strategies and accurate expectations of failover times in actual failover situations.

B.1.1.1 Full Client Startup and the CDS Server

When the CDS server is unavailable, full client startup does not represent a true failover scenario. Upon startup, a full client writes to the master CDS server to provide its location. If the client cannot contact the master CDS server, it will try to use the backup CDS server. However, it does not truly fail over to the backup CDS server.

Full client startup writes to CDS only if the full client configuration is different from the previous full client configuration (such as when a DHCP server assigns it a different IP address).

Rather, the client continues trying to contact the master CDS server, and has partial functionality during this time. If the client’s IP address is unchanged (this may not be the case if you are running DHCP), the client should still be able to perform most operations, such as status-type operations like **dcecp server ping**. However, the client will be unable to run applications that need to write to the CDS namespace.

B.1.1.2 Interaction Between Application and Servers

If you are using a DCE application when the CDS server goes down, this is not a failover scenario. If an application session is in progress, the client has already obtained the application server’s bindings from the CDS server. At this point, if the CDS server fails, the application session can still continue.

If you are using an application over the network when the security server goes down, you can continue to use the application while your credentials are good (by default, credentials are good for two hours at a time).

If the master security server is unavailable, failover does occur halfway through the credentials expiration interval, when the client contacts the security server to refresh the credentials. This process occurs automatically and is invisible to the user, unless a backup security server cannot be found.

Once your credentials expire, any process for which you require tickets will fail. When a client makes a request of an application server, the application server contacts the security server to see if the client’s credentials are sufficient to fulfill the request. This is also a failover scenario.

B.2 Requirements for Successful Failover

To prepare for successful client failover, you must ensure that replicas are maintained for security servers and CDS servers (see *Section 5.4 on page 69* for information about creating replicas).

By default, a backup CDS server replicates only the root directory; you should change the default replica configuration to maintain all of the additional directories that will be needed if the master is unavailable. Once you create directories on your backup server for all directories on the master, the backup directories are automatically synchronized on a periodic basis (the default is once per hour).

See *Section 5.5 on page 70* for instructions on changing the default CDS replica configuration.

B.3 Failover Test Environment

Review the information in this section for general expectations about failover times. Then, refer to *Section B.5 on page 126* to see if failover in your environment is subject to any additional factors.

The failover statistics in this section were obtained under the following test conditions:

- Windows NT 4.0 environment with Service Pack 3.
- Servers and replicas local to the network.
- One system housing the preferred CDS server and master security server, and another system housing the replica CDS and security servers.
- Registry keys set to their defaults.
- **RPC_SUPPORTED_PROTSEQS** environment variable used to set the environment to TCP-only for TCP readings and UDP-only for UDP readings.

The information in the CDS cache and in the **pe_site** file has a major impact on whether failover is required for PC-DCE operations, and on the speed of failover (see *Section B.5.1 on page 127* and *Section B.5.2 on page 127*). The tests in this section indicate failover readings when these files either do not exist or are not current.

B.4 Failover Test Results

The following sections include test statistics for the failover conditions described in *Section B.3*:

B.4.1 Failover when the CDS or Security Server is Not Running

These tests apply to scenarios in which the preferred CDS server or master security server is not running, but the system that houses them is up and running.

In either a TCP or UDP environment, failover to the backup servers for the following scenarios is immediate (within a few seconds).

- Client **dce_login**
- CDS lookup
- Refresh of client tickets (credentials) during an application session

Rapid failover occurs because the Windows 2000 or Windows NT endpoint mapper on the master server system immediately informs the client that the server is unavailable. The client does not need to wait for the protocol timeout period before contacting the backup server system.

B.4.2 Failover when the Server System is Unreachable

These tests apply to scenarios in which the client is unable to contact the system that houses the preferred CDS and master security servers (for example, the system is disconnected from the network or has been powered off). Under these circumstances, the endpoint mapper is not running, so failover is not immediate.

B.4.2.1 Client Performs a CDS Lookup

If the preferred CDS server is unreachable when the client needs to perform a lookup (for example, a **dcecp cell show** command or an attempt to locate an application server):

- **TCP** — Failover to the backup CDS server in a TCP environment takes approximately 1 minute 30 seconds.

  The client then caches the backup CDS server as master. Should the new master become unreachable, failover to the original master may take up to 5 minutes, depending upon the TCP state of the connection.

- **UDP** — Under UDP, failover to the backup CDS server takes approximately 3 minutes 40 seconds.

B.4.2.2 Client Logs into DCE

Upon login, the client runtime uses CDS to locate the security server, then refers to the **pe_site** file, *unless* you've configured the environment to use **pe_site** exclusively. See *Section B.5.2*.

If you are not using the **pe_site** file exclusively, failover to log into the backup security server takes:

- **TCP —** Approximately 2 minutes 15 seconds.
- **UDP —** Approximately 1 minute 13 seconds.

B.5 Factors that Affect Failover

Environmental factors include:

B.5.5 Replicas Across a WAN Link

Client failover for application servers is affected by a different set of factors, and is discussed separately in *Section B.6 on page 128*.

B.5.1 Cache Contents

CDS lookups are affected by whether or not the client has already stored an application server's bindings in its cache. For example, if an application session is in progress, the client has already obtained the application server's bindings from the CDS server. At this point, if the CDS server fails, the application session can still continue.

B.5.2 PE_Site File Use

The **pe_site** file is a list of security servers and their associated bindings. By default, the client runtime looks up bindings for a security server by using CDS. If the preferred CDS server is unavailable, the runtime will look up the location of a security server in the **pe_site** file.

If the **pe_site** file is up-to-date, a **dce_login** will succeed immediately although the security server may be down, because the information will be obtained from the **pe_site** file without needing to contact the security server.

You can force the runtime to use **pe_site** exclusively, rather than CDS. If you do so, an unavailable CDS server will never be an issue for client login, because the client runtime will never contact CDS. It will use the security server bindings already stored in **pe_site**. This could save time in the event that a preferred CDS server is unavailable (as long as the **pe_site** file is accurate).

Keep in mind that, although the **dce_update** process monitors server status and sorts available servers to the top of the **pe_site** list, failover will take longer when **pe_site** includes multiple servers, and the first available server is not towards the top of the list.

Refer to the *PC-DCE Overview Guide* for more information about the **pe_site** file.

B.5.3 Endpoint Mappers

The presence of an operating system endpoint mapping service in addition to PC-DCE’s endpoint mapping service has a major impact on the speed of client failover. When an operating system endpoint mapping service is in use on the server system, the client contacts the server and the endpoint mapper immediately responds that the server is down. This allows the client to quickly fail over to the replica server.

Windows 2000 and Windows NT 4.0 include their own endpoint mapper, so if you are running either of these sytems you can expect a faster failover than if you are running some of the UNIX operating systems, such as AIX or Solaris.

When no endpoint mapping service is in use, and in the event that PC-DCE is down, the protocol timeout period must pass before the client knows it must move on to a replica. For TCP, the timeout period is two minutes; for UDP, the timeout period is 45 seconds.

B.5.4 Registry Keys

Modifications that you make to registry keys can affect client failover time. For example, you can:

- Change the time period that PC-DCE waits for the security server to initialize before failing over to another security server (SecdWaitTimeout).
- Change the time period that PC-DCE waits for the CDS server to initialize before failing over to another CDS server (CdsdWaitTimeout).
- Increase the frequency at which **dce_update** solicits and caches CDS clearinghouses and security server information (CDSUpdateInterval, SECUpdateInterval). The more frequent the solicitations, the fresher the cache. However, note that frequent solicitations incur more network traffic.

For information about modifying registry keys, see *Appendix A on page 117*.

B.5.5 Replicas Across a WAN Link

Failover to a replica located across a WAN link is subject to the additional delays that may be incurred by the WAN link (for example, the link may have slower response times or bottleneck conditions).

B.6 Application Server Failover

In order for failover to occur for application servers, you must ensure that more than one application server is available to the client.

B.7 Responding to Loss of Service

If a primary server is permanently unavailable, then you must take steps to create a new primary server. DCE does not automatically create a new primary server. *Section 5.5.4.3* describes how to reconfigure a CDS backup server as a primary server, and *Section 5.6.1* describes how to promote a backup security server to master server.

If a master security server goes down, promote the backup server to master as soon as possible. It is not possible to write to backup security servers, so processes such as changing passwords can not occur until the server is promoted to master or the original master comes back up.

If your credentials expire before promoting the backup to master, when the original master comes back online you may not be able to log in to the new master to return it to backup status.

Index