ENTEGRITY *Solutions* ®

PC-DCE™

Overview Guide

Software Version 5.0

Notices

Contents

Preface

Intended Audience

This guide is for users who have a general knowledge of The Open Group (formerly Open Software Foundation) Distributed Computing Environment (DCE), and who want to learn more about the Entegrity® DCE implementation for Windows® called PC-DCE.

Contacting Entegrity Solutions

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
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

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**.

Obtaining Technical Support

If you purchased PC-DCE directly from Entegrity Solutions, 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: +1 781-376-9358 **http://www.opengroup.org**

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:

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

Documentation on other Entegrity products 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* (this book)
- *PC-DCE Administrator's Guide*
- *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 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*

CHAPTER 1

PC-DCE Overview

1

Entegrity® PC-DCE™ is a Windows implementation of OSF DCE Version 1.2.2, and is fully compliant with the The Open Software Foundation (OSF) DCE standards.

NOTE: Throughout documents related to Entegrity PC-DCE, use of the term Windows refers to all supported Windows operating systems unless noted otherwise.

The purpose of this document is to provide DCE developers and administrators an overview of how PC-DCE is implemented under Windows, special features that are available only with PC-DCE, and technical details regarding the client implementation.

This chapter contains the following sections:

1.1 OSF DCE Overview

OSF Distributed Computing Environment provides services and tools that support the creation, use, and maintenance of distributed applications in a heterogeneous computing environment.

By distributed computing we mean computing that involves the cooperation of two or more machines communicating over a network. The machines participating in the system can range from personal computers to supercomputers; the network can connect machines in one building or on different continents.

1.2 How PC-DCE Implements DCE Under Windows

This section discusses how PC-DCE implements specific aspects of OSF DCE in a Windows environment.

1.2.1 Client/Server Architecture

A PC-DCE cell includes client systems and at least one system installed with server components. Each cell member running Windows must have the PC-DCE Client Runtime installed. The runtime component is included in both the Server kit and the Client Runtime kit.

1.2.1.1 Server

The PC-DCE server implementation integrates DCE service daemons directly into the Windows Services subsystem. The primary daemons are:

- Security Service daemon (**secd**)
- Cell Directory Service daemon (**cdsd**)
- Cell Directory Service Advertiser (**cdsadv**)
- DCE daemon (**dced**)
- Distributed Time Service daemon (**dtsd**)
- Global Directory Agent (**gdad**)
- Name Service Interfaces daemon (**nsid**)

NOTE: The **nsid** can also function as a client.

The implementation of the client components under Windows is treated in detail in *Chapter 2*.

1.2.2 DCE_Service Process

The **dce_service** process manages a number of different tasks within your DCE environment. The actual list of services it ultimately offers depends on the type of configuration you choose.

The **dce_service** process also performs the following housekeeping functions:

- Deletes stale credential files

- Manages the **dce_update** process

- Starts, stops, and displays individual DCE processes in a server or full client configuration

- Handles the priming mechanism for the optional Endpoint Services Only option

1.2.3 Endpoint Mappers

On all supported Windows operating systems except Windows 98, PC-DCE uses the Microsoft endpoint mapper service, **rpcss.exe**, to provide socket lookup services for applications. Enable Endpoint Service Only on the Options tab of the PC-DCE Configuration Panel to start the DCE endpoint primer, which determines if **rpcss.exe** is running, and starts it if it is not. See *Section 1.2.3 on page 10*.

On Windows 98, **dced.exe** provides endpoint mapping services.

1.2.4 Error Message Handling

On all supported Windows operating systems except Windows 98, all error messages generated by the DCE services are written directly to the Windows NT Event Logger to provide consistent error handling.

On Windows 98, DCE errors are logged to the *install_directory***dce32.log** file.

1.2.5 Environment

DCE environment information (environment variables, path information, login credentials, and so on.) is incorporated into the Windows Registry and system environment variables. You can modify registry keys and environment variables to fine-tune PC-DCE behavior.

1.2.6 Integrated Login

PC-DCE provides integrated login to Windows and to DCE. When a user logs into Windows, PC-DCE automatically logs the user into DCE. For integrated login to work:

■ The user's Windows and DCE user names must be identical
■ The user's Windows and DCE passwords must be identical
■ The integrated login feature must be enabled in the local PC-DCE configuration

The integrated login feature also detects when the user changes his Windows password and automatically updates the DCE password.

1.2.7 Pthreads

PC-DCE offers the common DCE pthread API, but uses the available Windows kernel threads under that interface. This integration lets developers monitor pthreads using native Windows tools such as the Performance Monitor and the Process Viewer provided with the Win32® Software Developer's Kit and Resource Kit.

1.2.8 PC-DCE Tools

You can manage PC-DCE using a variety of graphical and command-line tools.

1.2.8.1 DCEsetup

DCEsetup, shown in *Figure 1-1*, is a tool that provides an easy-to-use graphical interface for configuring DCE services on your Windows system. With DCEsetup, you can configure the following DCE components:

■ DCE Security Service
■ DCE Cell Directory Service (CDS), including:
    ■ Name Service Interface Gateway

- ■ Global Directory Agent (GDA)
- ■ Distributed Time Service (DTS)

DCEsetup can configure these components so that your Windows NT or Windows 2000 system can function as a:

- ■ DCE client system
- ■ Full DCE server system
- ■ Split server system
- ■ CDS read-only replica server
- ■ Security read-only replica server

Figure 1-1: DCEsetup
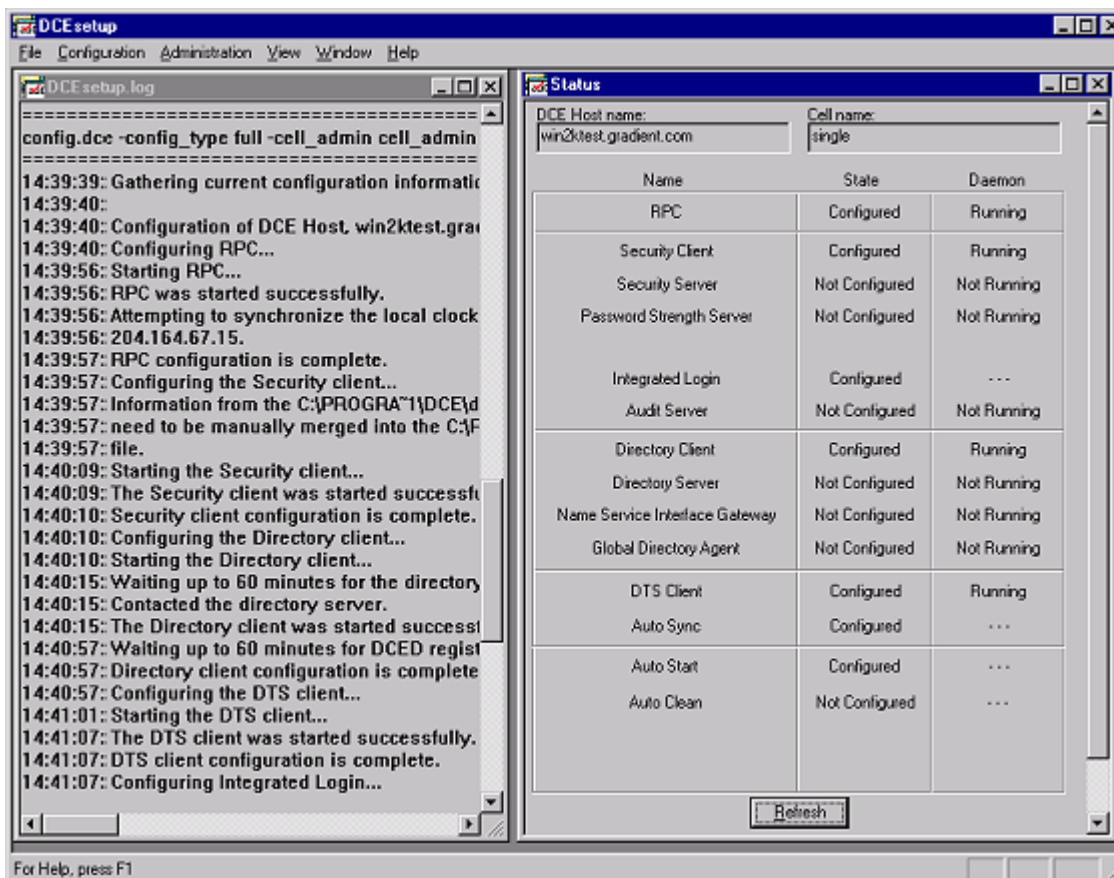
When you configure a system as a server, you automatically configure DCE client services on that system as well.

Many of the text entry fields that you will encounter during configuration have default values associated with them. These default values are based on your existing configuration, if you have one. Otherwise, DCEsetup provides values that are appropriate for the most common DCE configurations.

You must be logged in as a member of the Windows Administrators Group to perform a DCE configuration or make changes to a configuration.

For more information on using DCEsetup, refer to the DCEsetup online help system.

1.2.8.2 DCE Director

The DCE Director is a graphical tool for managing DCE cells. The DCE Director (*Figure 1-2*) 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 (see *Section 1.2.8.3 on page 14*), which allows you to graphically manage ACLs. You can invoke the Visual 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.

Figure 1-2: DCE Director
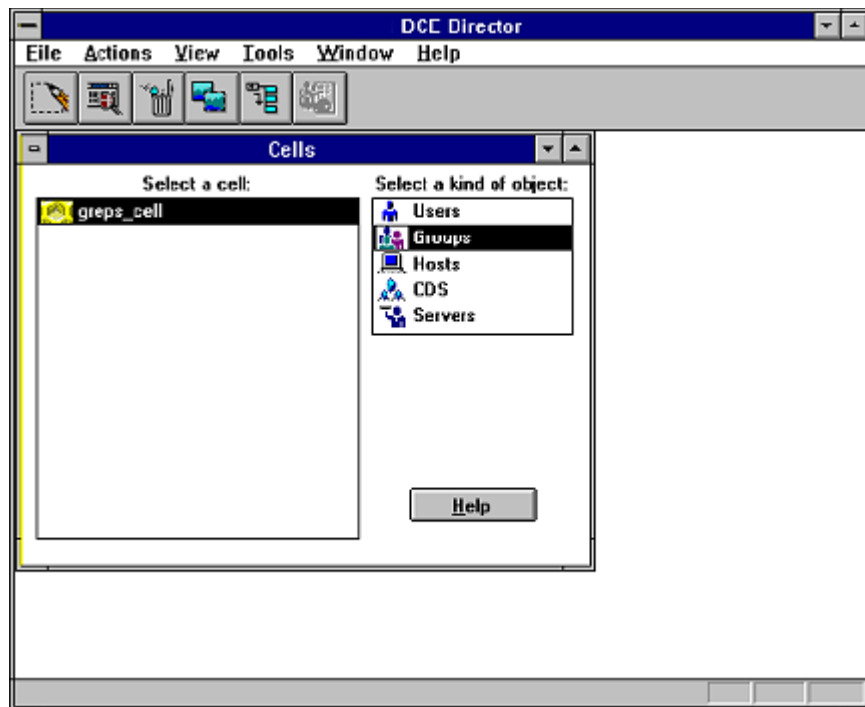
For more information on either the DCE Director or the Visual DCE ACL Editor, refer to their respective online help systems.

If you are not using DCE Director, you can use the DCE command line tools as described in *PC-DCE Administrator's Guide* to perform the same functions.

1.2.8.3 Visual DCE ACL Editor

The Visual DCE ACL Editor (*Figure 1-3*) makes it easy for you to set the permissions for all security-relevant objects within DCE, including Registry objects and CDS objects. It allows you to display, add, modify, copy, and remove ACL entries for a specific object in a cell's namespace. You can also go past junctions to application-specific namespaces and set permissions on the ACLs of application-specific objects.

Figure 1-3: Visual DCE ACL Editor
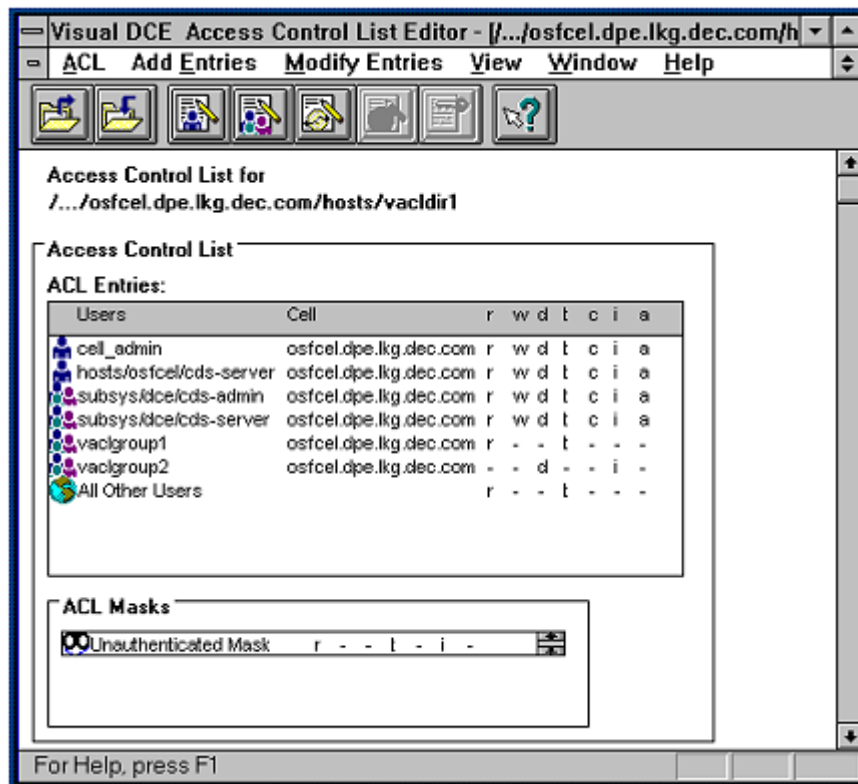
Among the tasks you can perform using the ACL Editor are the following:

■   View an object's ACL.

■   Create, modify, and remove ACL entries (from your home cell and in other cells).

■   Display the permissions implemented for an object's ACL by the object's application server.

■   Ask about your access to an object.

■   Ask about someone else's access to an object.

■   Copy one ACL to another ACL.

■   Create, modify, and remove masks used to restrict allowable permissions.

For more information on the Visual DCE ACL Editor, refer to the Visual DCE ACL Editor online help system.

1.2.8.4 PC-DCE Service Panel

All DCE services configured on your system are, by default, started automatically whenever you reboot your system. Sometimes, however, you may need to stop or restart PC-DCE manually.

On all Windows platforms supported, you can use the graphical PC-DCE Service Panel (*Figure 1-4*) to start, stop, test and review the operational status of individual DCE components.

Figure 1-4: PC-DCE Service Panel
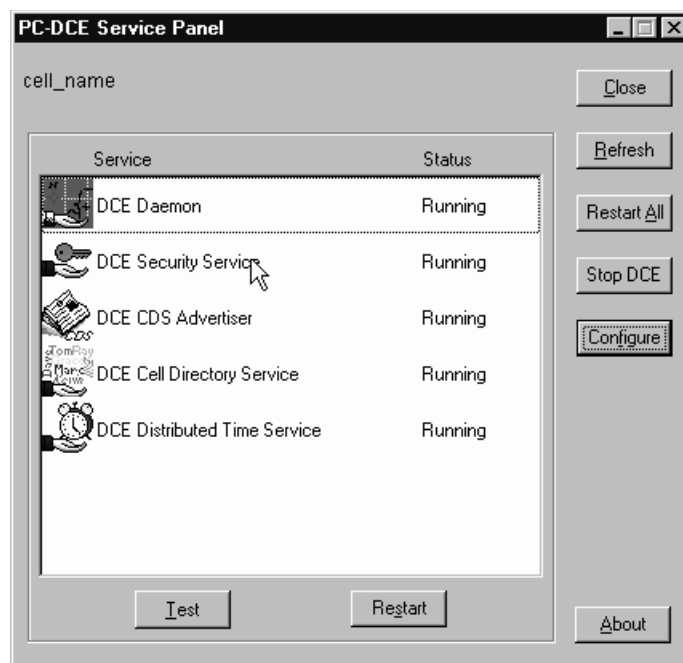
NOTE: On all supported Windows operating systems except Windows 98, you can also use the Windows Services control panel to stop and restart PC-DCE.

1.2.8.5 PC-DCE Configuration Panel

The PC-DCE Configuration panel (*Figure 1-5*) provides an additional tool for configuring DCE cells, servers, and clients.

Figure 1-5: PC-DCE Configuration Panel
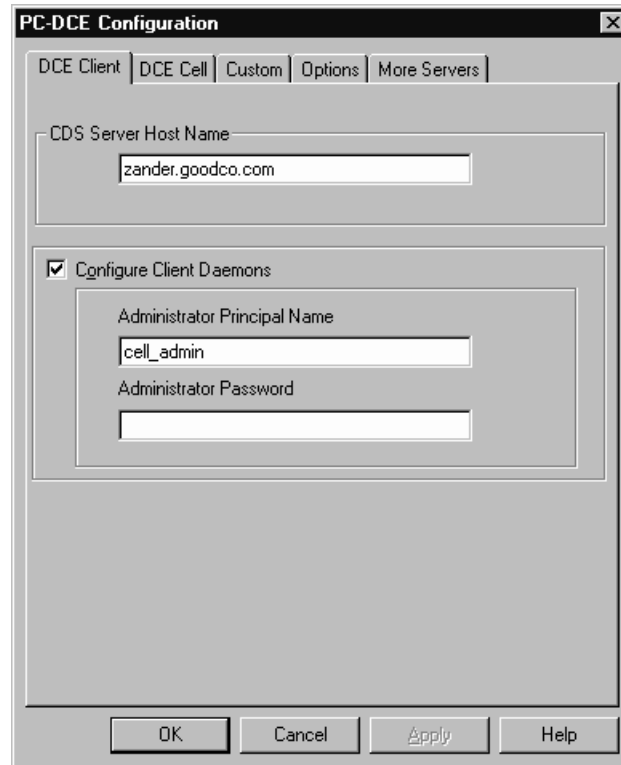
1.2.8.6 DCE Command Line Tools

PC-DCE fully implements **dcecp** (Distributed Computing Environment Control Program), the primary command-line management interface for managing DCE. **dcecp** allows you to manage core DCE administrative functions and administer DCE components remotely.

1.3 Special Features

In addition to the standard DCE functionality in The Open Group's DCE, PC-DCE provides some added features.

1.3.1 Lightweight Client

You can choose to configure a lightweight DCE client, which does not configure the **dced**, **dtsd** or **cdsadv** client daemons. This lightweight configuration reduces computing overhead on the client system and eliminates the need for you to specify the cell administrator principal and password during the configuration process. The lightweight client is discussed in detail in *Chapter 2*.

1.3.2 Per-Thread Login Contexts

Standard DCE allows you to have as many login contexts as you want, since you supply the login context handle when annotating an RPC binding handle with security or when negotiating a GSSAPI session. However, DCE allows only one process-wide default login context to be set via the **sec_login_set_context()** call.

PC-DCE enhances standard DCE to allow default login contexts on a per-thread basis. Your application calls **sec_login_set_thread_context()** to set up the thread-specific context. Then, calling **sec_login_get_current_context()** from that thread returns the per-thread context rather than the process-wide context.

1.3.3 CDS Preferencing

CDS preferencing lets you assign ranks to clearinghouses in a *preference file*, which PC-DCE reads at startup. In this way, you control a client's preference for CDS clearinghouses. CDS preferencing is useful in situations where multiple clearinghouses exist; if some of the clearinghouses are connected to the client's LAN by a low-performance WAN link, you can assign preference to a local clearinghouse.

1.3.4 Co-Authentication Service

The PC-DCE Co-Authentication Service (CAS) provides developers with the ability to plug alternative authentication methods into PC-DCE. A user logging in through CAS uses an alternative authentication method, for example a biometric device such as a fingerprint scanner, to obtain DCE login credentials.

1.3.5 C++ Support

PC-DCE includes an enhanced IDL compiler that supports C++ based application development. Developers can write client and server programs that access C++ objects transparently, independent of their location. C++ features such as inheritance and object references are supported.

1.3.6 Microsoft Terminal Server

PC-DCE v4.1 provides support for systems running Windows NT Terminal Server and Windows 2000 Terminal Server. This support is only available when you purchase PC-DCE specificlly to run in a Terminal Server Environment. The licence and cofiguration need to account for multiple clients configured with the Terminal Server. Special requirements at installation are explaind in the *PC-DCE Installation and Release Notes*.

1.3.7 Designating a Local Configuration Administrator

The **preconfig.tcl** script has been updated to include NetCrusader/Web support.

This script is a TCL/TK script that allows the cell administrator to designate a local administrator who can configure a DCE client on host machines. This allows local administrators to configure a full client into a cell without knowing the cell administrator password.

Designation can be by principal name or group name. If you designate a group, any member of the group can be a local administrator.

This feature is sometimes referred to as a preconfiguration script because the script does some of the configuration ahead of time, before the local administrator finishes the configuration. This feature is also sometimes referred to as a split configuration because the cell administrator uses the script to perform some of the configuration and then splits off the rest of the configuration tasks to a local administrator, who performs further administrator tasks on the remote client host.

NOTE: 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.

Information You Need to Run preconfig.tcl

The **preconfig.tcl** script asks for the following information:

- A list of fully qualified domain names of the hosts that you want to configure

- An entity name (principal or group name)

- The entity type for the entity name (group or principal)

- Administrator password

The entity (single or group) name must have a valid login in the cell to which the client machine will belong.

What the Script Does

The script creates new groups, accounts, and CDS entries necessary for the client to perform post configuration tasks such as adding the client host to the cell.

Installation steps are in the *PC-DCE Administrator's Guide* section 2.3.1. See the *PC-DCE Configuration Panel* help file for information about configuring clients.

1.4 PC-DCE Kit Components

PC-DCE kits are available in domestic and export versions. Due to U.S. federal trade restrictions, encrypted RPC (packet privacy) and encryption via the GSSAPI are disabled in the export kit.

1.4.1 Client Runtime Kit

The PC-DCE Runtime Kit (RTK) includes the PC-DCE client runtime. Once you install the Runtime Kit, you can configure the system as a DCE client in an existing cell. The client runtime runs on all Windows platforms supported.

The Client Runtime Kit includes the PC-DCE runtime software, the message catalogs, sample client/server programs, and basic administration utilities.

1.4.2 Server Kit

PC-DCE Server Kit includes the client runtime, CDS server, and standard security server.

1.4.3 Application Developer's Kit

PC-DCE Application Developer's Kit (ADK) includes the libraries, utilities, and header files that you need to create DCE-compliant applications for the Windows environment.

CHAPTER 2

Client Configurations

2

This chapter provides a deeper level of technical detail on the NetCrusader DCE/PC-DCE client runtime implementation. The primary reason for this information is to provide developers and administrators with a sufficient understanding of the *full* and *lightweight* client options offered by NetCrusader DCE/PC-DCE.

This chapter contains the following sections:

2.1 Client Component Overview

Figure 2-1 and the sections that follow describe the main components of a client configuration. For the purpose of discussion, the full client configuration is shown; the lightweight client configuration does not include **dced** and **cdsadv**.

Figure 2-1: PC-DCE Client Configuration
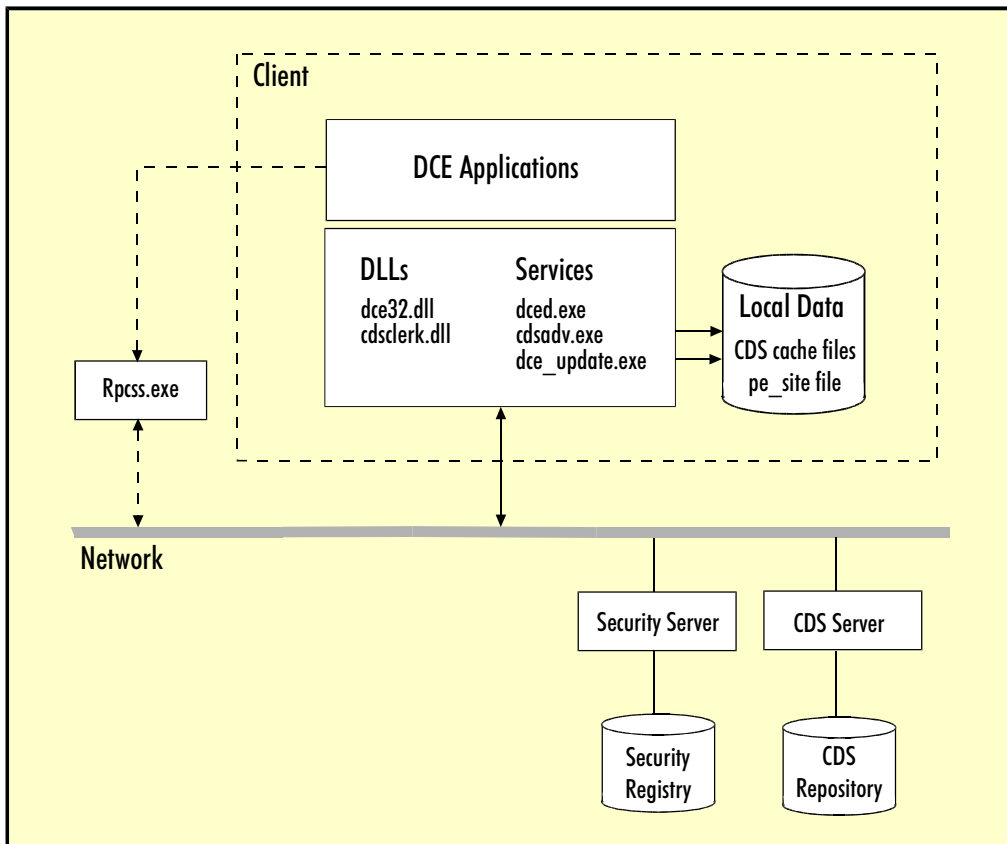
2.1.1 cdsadv.exe

The CDS Advertiser (**cdsadv.exe)** assists the CDS clerk in maintaining the local CDS cache files, which DCE clients use when trying to locate network resources. The CDS Advertiser monitors the LAN for broadcasts from CDS servers, which contain information such as server status and clearinghouse name. The CDS Advertiser populates the CDS cache with this information, entering new servers and updating server status.

2.1.2 dced.exe

The dced daemon, **dced.exe**, supports the following services:

- On all supported Windows operating systems except Windows 98, the Microsoft endpoint mapper is used rather than the **dced** endpoint mapper.

  Local endpoint mapping services (Windows 98 only).

- Third-party pre-authenticated logins, which may or may not be required depending on the security requirements at your site.

- Certification of credentials, which proves that the credentials were not obtained from a rogue security server. Again, these may or may not be required depending on the security requirements at your site.

- Machine credential creation and maintenance.

2.1.3 dce32.dll

dce32.dll includes the DCE runtime.

If a DCE application needs to call the security server, it uses the security APIs in the runtime. If a DCE application needs to contact the CDS namespace, it calls the RPC routines in the runtime; the runtime then calls the routines in the CDS clerk to get the namespace information.

2.1.4 CDS Clerk

In UNIX-based DCE, the CDS clerk is implemented as a process. In PC-DCE, the CDS clerk is implemented as a Windows DLL (**cdsclerk.dll**). The CDS clerk includes the CDS runtime.

Any application that needs access to CDS must go through the CDS clerk. Applications that request access to the CDS namespace call RPC routines in the DCE runtime; the DCE runtime then calls routines in the CDS clerk to obtain the requested information from CDS.

The CDS clerk, along with certain application clients and the CDS Advertiser, maintain the CDS cache. The CDS clerk queries CDS servers for namespace information on behalf of the client. The clerk also helps maintain cache records of which servers are currently responding and which are not.

2.1.5 CDS Cache

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.

The CDS cache is maintained in two areas:

- Common (global) cache, which contains information available to everyone (directory entries, clearinghouses, and so on).

- Per-user cache, which contains user-specific information (softlinks, groups, directory entries).

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

■ The lifetime (roughly 8-10 hours) of the cached entry expires. (The exception to this lifetime is CDS server entries, which are updated hourly by **dce_update**.) Expired entries are updated when the clerk attempts to use the information and finds it to be inaccurate.

■ The user establishes new credentials (probably through a new **dce_login**). This updates both the common and per-user cache files.

The PC-DCE configuration process creates the initial CDS cache by reading the root directory of the client's primary CDS server, whose name is supplied by the person performing the configuration. The root directory contains references to all of the CDS clearinghouses and their associated CDS servers.

The CDS cache is maintained primarily by the CDS clerk, with help from the CDS Advertiser and certain application clients.

■ CDS Clerk maintenance — When the CDS clerk receives a CDS servers'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.

■ CDS Advertiser maintenance — The CDS Advertiser monitors the LAN for CDS server broadcasts. These broadcasts inform the CDS Advertiser about new CDS servers, which it enters into the cache. CDS Advertiser also uses the broadcasts to update cache records of which servers are currently responding.

2.1.6 dce_update

dce_update.exe is a lightweight PC-DCE process that keeps the CDS cache and **pe_site** file up-to-date.

Each clearinghouse entry in the CDS cache is designated as *OK* or *Not OK*. Every hour (this period is tunable), the **dce_update** process solicits clearinghouses that are marked Not OK, and changes the setting to OK if it is now responding.

This is especially important if a preferred server outside the LAN goes down. Since the server is outside the LAN, no broadcast occurs to indicate when it is available again. **dce_update** monitors the server's status, ensuring that the client returns to it after the preferred server comes back on line.

The **pe_site** file contains a list of security servers and associated bindings that the DCE runtime uses to select a security server. Every hour (this period is tunable), the **dce_update** process pings all known security servers and moves servers that do not respond to the bottom of the **pe_site** list, keeping the list sorted so that available servers are listed first.

The **dce_update** process checks for any registry keys or variables that you have modified, such as SEC_DEFAULT_ENTRY, which you can use to specify a preferred security server. These modifications are taken into account when **dce_update** creates the **pe_site** file.

2.2 Understanding Lightweight Clients

By default, the PC-DCE Configuration program configures your system as a lightweight DCE client, which does not include configuration of the **dced**, **dtsd** or **cdsadv** client daemons. Lightweight clients participate as a functioning member of a cell while offering performance and overhead advantages that make them preferable to full clients.

The information in this section will help you decide if a lightweight client configuration will be effective in your environment.

2.2.1 Lightweight Versus Full Clients

Configure lightweight clients to minimize the amount of memory PC-DCE uses, or when you want someone who is not the cell administrator to be able to manage his or her own PC-DCE configuration.

Lightweight clients are different from full clients in the following ways:

■ Lightweight clients do not run **dced**. In a *full* client configuration, at startup, **dced** must register its bindings with the master CDS server. This involves approximately 180 Kbytes of network traffic. This is a disadvantage in a large cell or when the traffic must span a WAN link. **dced** consumes local CPU resources, and its startup requirements slow down system reboots.

By running without **dced**, lightweight clients save space in CDS clearinghouses because no entries are required in the **hosts** directory. Space is also saved in the security registry because no machine principal accounts are required.

The lightweight client compensates for the absence of **dced** by offering an optional endpoint mapper service that lets you run local applications (see *Section 1.2.3 on page 10*).

■ Lightweight clients do not run **cdsadv**. In a *full* client configuration, at startup, **cdsadv** must register its bindings with the master CDS server. This involves approximately 70 Kbytes of network traffic, which is a disadvantage in a large cell or when the traffic must span a WAN link.

cdsadv consumes local CPU resources, and its startup requirements slow down system reboots.

The lightweight client compensates for the absence of **cdsadv** by using the **dce_update** process, which periodically tests each server marked as not responsive in the CDS cache, and updates the cache if the server is now responding.

The benefits of the lightweight configuration are:

■ The person who configures the lightweight configuration does not require the cell administrator's account name and password.

■ A lightweight client does not require entries in the CDS **hosts** directory, conserving CDS resources as well as expediting lookups and the replication process. This advantage becomes very significant in larger cells.

■ A lightweight client does not require a registry entry for the *machine*/**self** principal, conserving space in the registry.

■ The lightweight client configuration process does not make changes to the CDS namespace or the security registry, so it completes quickly.

■ Because a lightweight client does not run daemon processes, it uses less local computing resources.

■ Because a lightweight client avoids daemon startup overhead, the local machine comes up faster.

■ Because daemons must register their bindings at startup, a lightweight client uses less local network resources.

2.2.2 Conditions That Require a Full Client

Depending upon the requirements at your site, you may need to run a full client instead of a lightweight client:

■ Lightweight client configurations do not generate machine and host self credential files, which are required by certain software programs (such as older versions of DFS). Check with your software vendor to determine if the software can run on a system configured as a lightweight client.

■ A lightweight client configuration does not support third-party pre-authenticated logins, which may or may not be required depending on the security requirements at your site.

■ A lightweight client configuration does not support certification of credentials. These may or may not be required depending on the security requirements at your site.

NOTE: In all supported Windows operating systems except Windows 98, if endpoint mapping services are required in a lightweight client configuration (if you want to run local server applications), you can configure PC-DCE to use **rpcss.exe**. *Section 1.2.3 on page 10*

If you are running Windows 98, you cannot have a lightweight client configuration and endpoint mapping services

---

Index

P

Path information   11
PC-DCE
    additional documentation   7
    configuration panel   15
    kits   18
pe_site file
    function   24
Permissions   14
Per-thread login contexts   17
preconfig.tcl   17
Pthreads   11

R

RPC
    encrypted   18
rpcss.exe   10
Runtime components   10
Runtime kit   19

S

Security Service daemon (secd)   10
Server components   10
Server kit   19
Split configuration   18
Support   5, 6

T

Technical support   5, 6
Terminal Server   17

V

Visual DCE ACL Editor   14