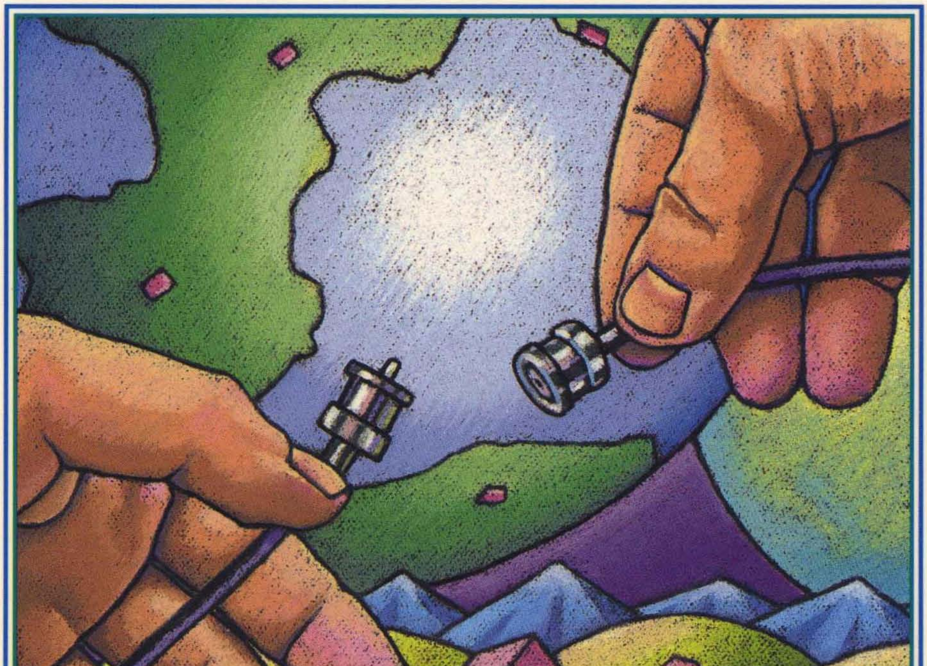
OSF™ DCE

OSF™ DCE Administration Guide — Introduction



OPEN SOFTWARE FOUNDATION

OSF® DCE Administration Guide – Introduction

Revision 1.0

Open Software Foundation

P T R Prentice Hall, Englewood Cliffs, New Jersey 07632

Cover design: **BETH FAGAN**
Cover illustration: **STEVE LEWONTIN**

This book was formatted with troff

Published by P T R Prentice-Hall, Inc.
A Simon & Schuster Company
Englewood Cliffs, New Jersey 07632

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

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

OSF 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.

Printed in the United States of America
10 9 8 7 6 5 4 3 2 1

ISBN 0-13-176546-9

Prentice-Hall International (UK) Limited, *London*
Prentice-Hall of Australia Pty. Limited, *Sydney*
Prentice-Hall Canada Inc., *Toronto*
Prentice-Hall Hispanoamericana, S.A., *Mexico*
Prentice-Hall of India Private Limited, *New Delhi*
Prentice-Hall of Japan, Inc., *Tokyo*
Simon & Schuster Asia Pte. Ltd., *Singapore*
Editora Prentice-Hall do Brasil, Ltda., *Rio de Janeiro*

Contents

Part 1. Introduction to DCE System Administration

Part 2. Configuring and Starting Up DCE

List of Figures

List of Tables

Preface

The *OSF DCE Administration Guide* provides concepts and procedures that enable you to manage the Distributed Computing Environment (DCE). Basic DCE terms are introduced throughout the *OSF DCE Administration Guide*. A glossary for all of the DCE documentation is provided in the *Introduction to OSF DCE.* The *Introduction to OSF DCE* helps you to gain a high-level understanding of the DCE technologies and describes the documentation set that supports DCE.

Audience

This guide is written for system and network administrators who have previously administered a UNIX environment.

Applicability

This is Revision 1.0 of this guide. It applies to the OSF® DCE Version 1.0 offering and related updates. (See your software license for details.)

Purpose

The purpose of this guide is to help system and network administrators to plan, configure, and manage DCE. After reading the guide, you will understand what the system administrator needs to do to plan for DCE. Once you have built the DCE source code on your system, use this guide to assist you in installing executable files and configuring DCE. The *OSF DCE Release Notes* contain instructions for installing and building DCE source code.

Document Usage

The *OSF DCE Administration Guide* consists of three books, each of which is divided into parts, as follows:

- *OSF DCE Administration Guide—Introduction*
  - — Part 1. Introduction to DCE Administration
  - — Part 2. Configuring and Starting Up DCE
- *OSF DCE Administration Guide—Core Components*
  - — Part 1. DCE Remote Procedure Call
  - — Part 2. DCE Cell Directory Service
  - — Part 3. DCE Distributed Time Service
  - — Part 4. DCE Security Service
- *OSF DCE Administration Guide—Extended Services*
  - — Part 1. DCE Global Directory Service

— Part 2. DCE Distributed File Service

— Part 3. DCE Diskless Support Service

Related Documents

For additional information about the DCE, refer to the following documents:

- *Introduction to OSF DCE*
- *OSF DCE Administration Reference*
- *OSF DCE User's Guide and Reference*
- *OSF DCE Release Notes*
- *OSF DCE Porting and Testing Guide*
- *OSF DCE Application Development Guide*
- *OSF DCE Application Development Reference*

Typographic and Keying Conventions

This guide uses the following typographic conventions:

Bold — **Bold** words or characters represent system elements that you must use literally, such as commands, options, and pathnames.

Italic — *Italic* words or characters represent variable values that you must supply.

`Constant width` — Examples and information that the system displays appear in `constant width` typeface.

[ ] — Brackets enclose optional items in format and syntax descriptions.

{ } — Braces enclose a list from which you must choose an item in format and syntax descriptions.

| A vertical bar separates items in a list of choices.

< > Angle brackets enclose the name of a key on the keyboard.

... Horizontal ellipsis points indicate that you can repeat the preceding item one or more times.

This guide uses the following keying conventions:

<Ctrl-*x***>** or ^*x* The notation **<Ctrl-***x***>** or ^*x* followed by the name of a key indicates a control character sequence. For example, **<Ctrl-c>** means that you hold down the control key while pressing **<c>**.

<Return> The notation **<Return>** refers to the key on your terminal or workstation that is labeled with the word Return or Enter, or with a left arrow.

Problem Reporting

If you have any problems with the software or documentation, please contact your software vendor's customer service department.

Pathnames of Directories and Files in DCE Documentation

For a list of the pathnames for directories and files referred to in this guide, see the *OSF DCE Administration Guide* and the *OSF DCE Porting and Testing Guide.*

Part 1

Introduction to DCE System Administration

Chapter 1

Introduction to DCE System Administration

This chapter introduces you to the major concepts that are needed for managing your OSF Distributed Computing Environment (DCE). These concepts are discussed throughout the *OSF DCE Administration Guide*.

DCE is a set of services that together make up a high-level, coherent environment for developing and running distributed applications. These services include a set of tools that support DCE management tasks. DCE applies techniques that you may have learned from working with applications for single machines or other distributed systems. These techniques enable system administrators to manage DCE without having to know about system internals. You can start with a configuration that is appropriate for your initial needs and grow to larger configurations without sacrificing reliability or flexibility. DCE supports large networks with many users, as well as smaller networks.

The following concepts, which are described in the remaining sections of this chapter, are central to DCE system administration:

- Clients and servers to make and respond to requests for a service
- Remote Procedure Calls (RPCs) for client-to-server communications
- Cells, which are groups of users, servers, and machines that share security, administrative, and naming boundaries

- A single namespace that lets client applications identify, locate, and manage objects, including users, machines, servers, groups of servers, and directories
- A single filespace that allows data sharing among users and machines that have proper authorization
- Principals, which are entities—including users, servers, and computers—that are capable of communicating securely with other entities
- Access Control Lists (ACLs) to control access to objects
- Caching, which is the technique of using a local copy of information to avoid looking up the centrally stored information each time it is needed
- Replication, which is the process by which copies of information are created and kept consistent

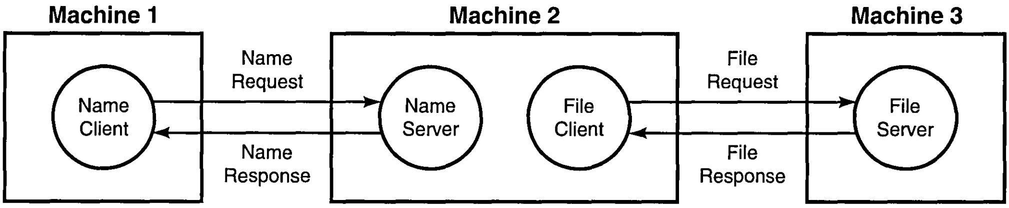
1.1 Clients and Servers

DCE is based on the client/server model. A server is a machine or process that provides a specialized service to other machines or processes. A client is a machine or process that uses a server's specialized service during the course of its own work. Distributed applications consist of a client side that initiates a request for service, and a server side that receives and executes that request, and returns any results to the client. For example, a client can request that a file be printed, and the server where the printer resides carries out that request.

More than one server process can reside on a single machine. Also, one machine can be both a client and a server. For example, a machine can be a client for one DCE component and a server for another.

Figure 1-1 shows a machine that is a name server for a client that issues a name request. The same machine is a client for a file server.

Figure 1–1. Interaction of Clients and Servers



1.2 Remote Procedure Call

A Remote Procedure Call (RPC) is a synchronous request and response between a local calling program and a remote procedure. An RPC begins with a request from a local calling program to use a remote procedure. It completes when the calling program receives all the results (or an error status or exception) from the procedure.

1.3 The Cell

The cell is the basic unit of administration in DCE. A cell usually consists of users, machines, and resources that share a common purpose and a greater level of trust with each other than with users, machines, and resources that are outside of the cell. Members of a cell are usually located in a common geographic area, but they can also be located in different buildings, different cities, or even different countries, provided they are adequately connected. A cell's size can range from only one machine to several thousand, depending on the size of the organization. All machines in an organization can be included in one cell, or you can choose to have numerous cells within one organization.

Cells designate security, administrative, and naming boundaries for users and resources. Each cell has a name. Cell names are established during the installation and configuration of DCE components.

Members of an organization who are working on the same project are likely to belong to the same cell. For example, in a large organization with several cells, the sales team could belong to one cell, the engineers working on Project X could belong to a second cell, and the engineers working on Project Y could belong to a third cell. On the other hand, a small organization may have only one cell for both the sales force and the engineers because they share the same level of security and the organization's small size does not warrant the additional administrative overhead that maintaining additional cells requires.

DCE services are managed within the context of a cell, as described by the following examples:

- Each DCE cell typically consists of at least two directory servers, three time servers, and one security server, as well as the databases that the directory and security servers use.
- The pathnames of DCE objects that are managed by DCE services can be expressed relative to the cell where the objects reside.
- The DCE Distributed Time Service (DTS) has both local and global servers. Local servers operate within a Local Area Network (LAN). Global servers provide time services anywhere within the cell.

1.4 The DCE Namespace

The DCE namespace is the hierarchical set of names of DCE objects. The top levels of the hierarchy are managed by the DCE Directory Service. Some DCE services (currently the DCE Security Service and the DCE Distributed File Service) manage their own portions of the DCE namespace. Each DCE object in the namespace consists of a name with associated *attributes* (pieces of information) that can be used to locate it. These objects include resources such as machines or applications.

The DCE namespace contains global namespaces and cell namespaces. A *cell namespace* includes objects that are registered within a cell. A logical picture of a cell namespace is a hierarchical tree with the cell root directory at the top and one or more levels of directories containing names beneath the cell root. The cell namespace is managed by the Cell Directory Service (CDS) component of the DCE Directory Service. Conversely, the *global namespace*, as seen from a local DCE cell, contains objects that are registered outside the local cell, such as the names of other cells. The Global Directory Service (GDS) component of the DCE Directory Service manages one part of the global namespace; a non-DCE service called the Domain Name System (DNS) manages another part of the global namespace.

Administrative tools use the namespace to store information and to locate DCE services. DCE services advertise their locations to the DCE namespace. The namespace provides a means of organizing DCE services into manageable groups.

1.5 The Filespace

Part of the DCE cell namespace is the filespace, which consists of files and directories. These can be physically stored on many different machines, but are available to users on every machine, as long as those users have the proper authorization. You manage the filespace in units called *filesets*, which are hierarchical groupings of related files. Although files are distributed throughout the network, located on and managed by different servers, users see a single filespace. DCE provides administrative tools to assist you in backing up, moving, and replicating filesets.

1.6 Principals

A DCE principal is an identity that is authenticated by the DCE Security Service. When you log into your system, you use your principal name. Principals can be organized into groups and into organizations that contain groups of principals. Information that is associated with a principal includes information that is traditionally kept in UNIX group and password files, such as the username, group ID, members of a group, and a user's home directory. By default, a principal is known within the bounds of a cell. By creating a special account that indicates you trust another cell's authentication service, you can enable principals from other cells to participate securely within your cell.

1.7 Access Control Lists

An Access Control List (ACL) is an authorization mechanism that allows you to assign permissions that control access to DCE objects. The following DCE objects are protected by ACLs:

- Principals and groups of principals that are managed by the DCE Security Service
- Files and file system directories that are managed by the DCE Distributed File Service (DFS)
- DTS servers
- CDS directories and entries
- CDS clients and servers, which have ACLs restricting the use of their management operations; for example, creating a clearinghouse
- GDS entries that are managed by GDS's own ACL mechanism, as described in the *OSF DCE Administration Guide—Extended Services*

An ACL consists of multiple *ACL entries* that define the following:

- Who can use an object
- What operations can be performed on the object

In the filespace, ACLs are an extension of the UNIX system's file protection model. Whereas UNIX file system permissions are limited to the protection of files and directories, DCE ACLs can also control access to other objects, such as individual database entries, objects that are registered in the cell namespace, and objects that are managed by applications. DCE provides the **acl_edit** command to help you administer ACLs on all DCE objects, except GDS objects, whose ACLs are managed separately.

1.8 Caching

Information that is acquired over the network (for example, using RPC) can be stored in a memory or disk cache on the local machine. This technique reduces network load and speeds up lookups of frequently needed data. For example, information about the DCE namespace and the DCE filespace is cached by DCE client machines.

Caching can be configured on a service-by-service basis. Different caching mechanisms are used for different components of DCE. Each component has configurable options to improve the performance of your installation.

1.9 Replication

Replication increases the availability of resources by having copies of the resource on several machines. For example, with replication you can make database updates on one machine and have them automatically made on other machines in the network. You can replicate data, move replicas, and control the frequency of updates. The DCE Security Service, CDS, GDS, and DFS all provide replication facilities that are customized for their particular applications.

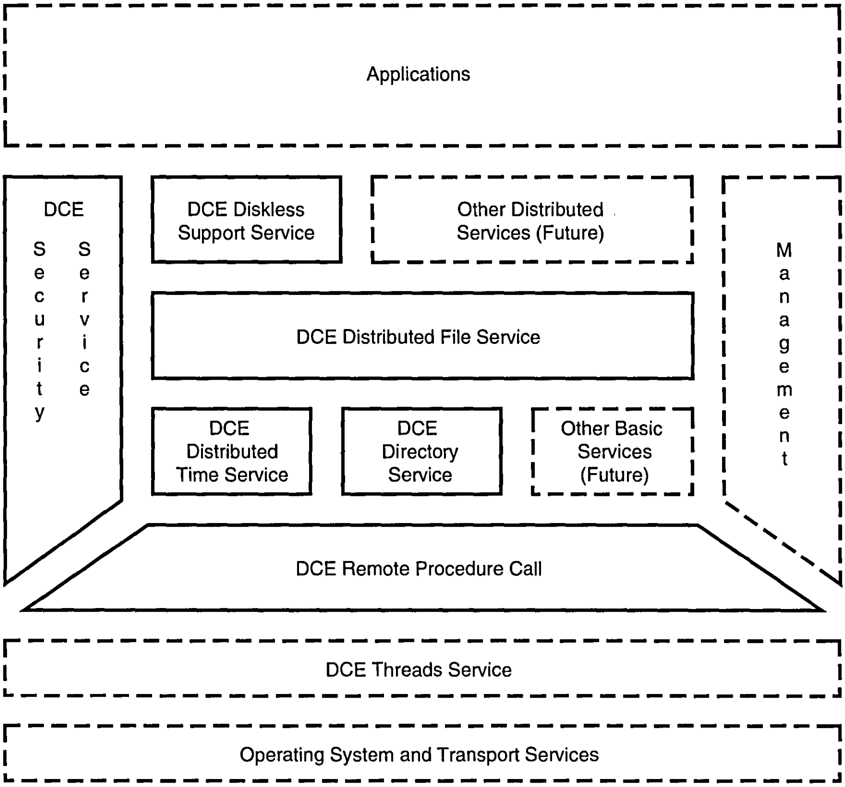
Chapter 2

Overview of DCE Components

This chapter provides an overview of the DCE components that are used by system administrators. This chapter also describes how components relate to each other, and it provides explanations of some key terms that are used in the DCE administration documentation. This chapter ends with a discussion of the range of tasks that make up system administration and where to find information about these tasks. (See the *Introduction to OSF DCE* for a discussion of all the DCE components and information about routines that application developers use.)

Figure 2-1 shows the different DCE components and how they fit together to form DCE. DCE resides between the applications shown at the top of the illustration and the operating system and transport services at the bottom. The boxes outlined with solid lines show components that comprise system administration functions that are discussed in this guide.

Figure 2–1. DCE Architecture


Applications
DCE Security Service
DCE Diskless Support Service
Other Distributed Services (Future)
Management
DCE Distributed File Service
DCE Distributed Time Service
DCE Directory Service
Other Basic Services (Future)
DCE Remote Procedure Call
DCE Threads Service
Operating System and Transport Services

2.1 Description of DCE Components

Although each of the DCE components serves a separate function, the components are interrelated. The following subsections describe the DCE components that require system administration.

The DCE Threads component of DCE is not discussed in the *OSF DCE Administration Guide* because there are no administrative tasks associated with threads.

The Management block shown in Figure 2-1 represents the administrative tools that assist you in managing DCE. Some of these tools are for general

DCE management, such as tools for configuring DCE. Others are for management of a specific DCE component, such as the program for managing the DCE Security database. These administrative tools are described in sections of this guide and in sections of the *OSF DCE Administration Guide—Core Components* and the *OSF DCE Administration Guide—Extended Services* that discuss individual DCE components and services.

2.1.1 DCE Remote Procedure Call

DCE Remote Procedure Call (RPC) is the primary method for client-to-server communications in DCE. The RPC daemon (**rpcd**) is a server process that supports the use of RPC services on a host. Every DCE machine must run **rpcd**. This includes DCE Client machines because DCE Clients also run processes that act as servers, such as **sec_clientd** and **dtsd**.

Each RPC-based server must register its addressing (or binding) information so that its clients can find it. The addressing information is typically comprised of two parts. The first part is the address of the machine on which the server runs. This information is stored in CDS, and gives enough information for a client to find the **rpcd** daemon on the server's machine, since the **rpcd** daemon has a well-known endpoint.

The second part of the server's address is the server's endpoint. (For the Internet Protocol (IP), an endpoint is a port.) The server gets a dynamically assigned endpoint when it starts up. The server must register its endpoint with **rpcd**; this is usually done during server initialization. The **rpcd** maintains RPC server information in a database that is known as the endpoint map. Once a client locates the **rpcd** on the server's machine, it can find out what the server's endpoint is from **rpcd** and then it can locate the server.

An administrative facility, the RPC control program (**rpccp**), can be used to manage both parts of the RPC binding information. The **rpccp** command allows you to do the following:

- Add and remove CDS entries that are specific to RPC bindings
- Add, retrieve, and remove information, such as binding information, from those entries

- Register, unregister, and show local and remote endpoint mappings

Refer to the *OSF DCE Administration Guide—Core Components* for more information about **rpcd** and **rpccp**. The *OSF DCE Application Development Guide* provides a detailed explanation of the remote procedure call model and DCE RPC.

2.1.2 The DCE Directory Service

The DCE Directory Service provides directory service at the cell and global levels. It allows both users and applications to store, retrieve, and manage information about objects such as computers, printers, users, and files. Because the DCE Directory Service facilitates the use of common naming conventions within a common namespace, users and applications are not restricted by physical location, brand of host system, or method of naming on a host system. Using common naming conventions allows sharing of information that is based on names, rather than location.

The DCE Directory Service stores attributes along with an object. An attribute is a piece of information that is associated with an object. An object's attributes can describe its class, network address, or other values. Therefore, a resource name does not need to change if it moves from one node to another. You can also search for a name given one or more of an object's attributes if the object is stored in the GDS part of the namespace.

The DCE Directory Service consists of the following components that have administrative functions:

- The Cell Directory Service (CDS)
- The Global Directory Service (GDS)
- The Global Directory Agent (GDA)

2.1.2.1 The DCE Cell Directory Service

CDS is the component that looks up and manages names within a cell. Client applications send their requests through a CDS clerk process, and if the data is not in a cache, it is passed to one or more servers to be handled. In a client configuration, one **cdsclerk** executable is installed on each machine, but several **cdsclerk** processes may be running at any one time. One CDS clerk is started for each DCE principal on a machine that makes CDS requests.

A CDS Server, the **cdsd** daemon, stores names and other CDS information in a database that is called a *clearinghouse*. The clearinghouse contains *replicas*, which are physical instances of CDS directories containing names.

A third process, the **cdsadv**, is responsible for sending and receiving advertisements of the presence of CDS Servers on DCE machines. The **cdsadv** process also spawns the CDS clerks that are needed on a machine and creates the cache that the CDS clerks share.

CDS administration tasks include the following:

- Configuring and replicating the CDS namespace
- Monitoring CDS Servers
- Managing access control on CDS directories

Once CDS is installed and configured, only occasional intervention for system administration is required. The following tools support CDS system administration tasks:

- The CDS control program (**cdscp**) is an interactive, command-line interface that you can use to configure the CDS namespace and perform maintenance tasks such as monitoring servers or creating a directory. With **cdscp** commands you can also display the structure and contents of the CDS namespace.
- The CDS Browser is a Motif-based application that allows you to display the overall structure of the CDS namespace, as well as view the contents of directories.

2.1.2.2 The DCE Global Directory Service

GDS, which conforms to the X.500 directory service standard, is the DCE component that supports the global naming environment outside of cells. GDS is somewhat independent of the DCE cell; for example, it does not use DCE RPC for interprocess communications. GDS maintains a directory that is used by DCE to store information about DCE cells. It can also be used as a general-purpose directory service.

The GDS directory is a distributed database. Each Directory System Agent (DSA), which is the server side of GDS, stores a different part of the database. A DSA can have copies of the information of other DSAs in order to increase availability and reduce response times. The original information is called a *master* and the copy is called a *shadow*. Every object in a DSA is either a master or a shadow. When an update occurs, usually the master object is modified. You can create jobs that periodically update shadows. The client side of a GDS configuration is known as a Directory User Agent (DUA).

GDS is administered through a menu-driven interface that allows you to do the following:

- Add, remove, and modify directory entries
- Replicate objects
- Monitor the directory structure
- Back up and restore local data files
- Activate and deactivate a directory system installation
- Alter the directory structure when necessary
- Manage access control lists

An important GDS administration task for the DCE system administrator is using the menu-driven interface to register DCE cell information. (See the *OSF DCE Administration Guide—Extended Services* for more detailed information about GDS administration.)

2.1.2.3 The GDA in Relation to the Cell and Global Directory Services

The third component of the DCE Directory Service is an independent process that is called the GDA. When CDS receives a directory request, it determines whether it can find the object in its own cell or whether it needs to contact another server or service to help it find the object. If the object is stored in another (foreign) cell, the CDS Server of the other cell must be called to resolve the name. To contact CDS in the foreign cell, the local CDS must know certain attributes of the foreign cell. CDS contacts the GDA to assist CDS in finding the attributes that are necessary to contact the other cell. Note that a GDA is only needed in a cell if communications with directory services in other cells are required.

Administration of the GDA consists of starting and stopping the GDA (the **gdad** process), and deciding how many **gdad** processes need to run in your cell. (See the *OSF DCE Administration Guide—Core Components* for more information on administering the GDA.)

Cell names and attributes can be registered in either GDS or another global directory service, which is called the Domain Name System (DNS). Many networks use DNS primarily as a name service for hostnames. Although DNS is not part of the DCE offering, it is supported by the DCE Directory Service and requires some administration.

The GDA determines whether the foreign cell object is in GDS or DNS, depending on the format of the cell name. GDS names are *typed*, consisting of a type and a value separated by an = (equal sign). If the GDA encounters a name such as **/C=US/O=ABCcompany**, it knows that the object belongs in GDS. If the name is untyped, consisting only of values such as **abc.com**, the GDA sends the request to DNS.

Figures 2-2 and 2-3 show a very simplified representation of how the GDA helps CDS resolve a name that is stored in GDS.

Figure 2–2. CDS Requests a Name in Another Cell

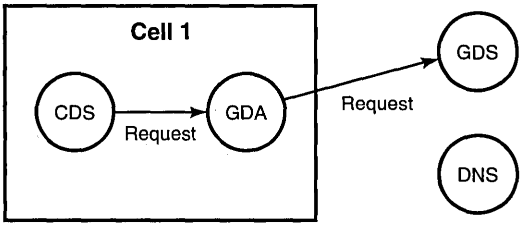
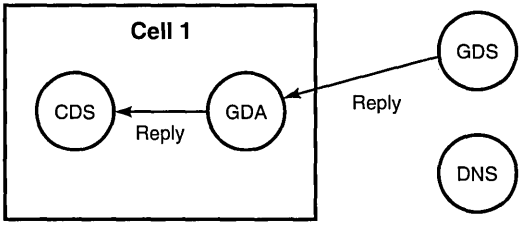
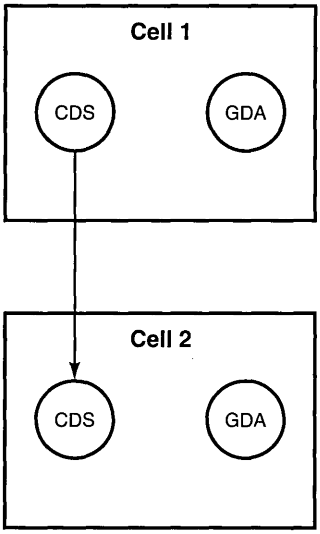


Figure 2–3. Location Information Is Returned to CDS



With this information, CDS can now directly access the other cell, as shown in Figure 2-4.

Figure 2–4. CDS Contacts Another Cell



To use both the local and global directory services that the DCE Directory Service provides, a cell must contain at least one CDS Server and at least one GDA. A cell can be configured to use CDS for local directory service and configured not to use the GDA, but users in this this cell are then unable to refer to objects in other cells.

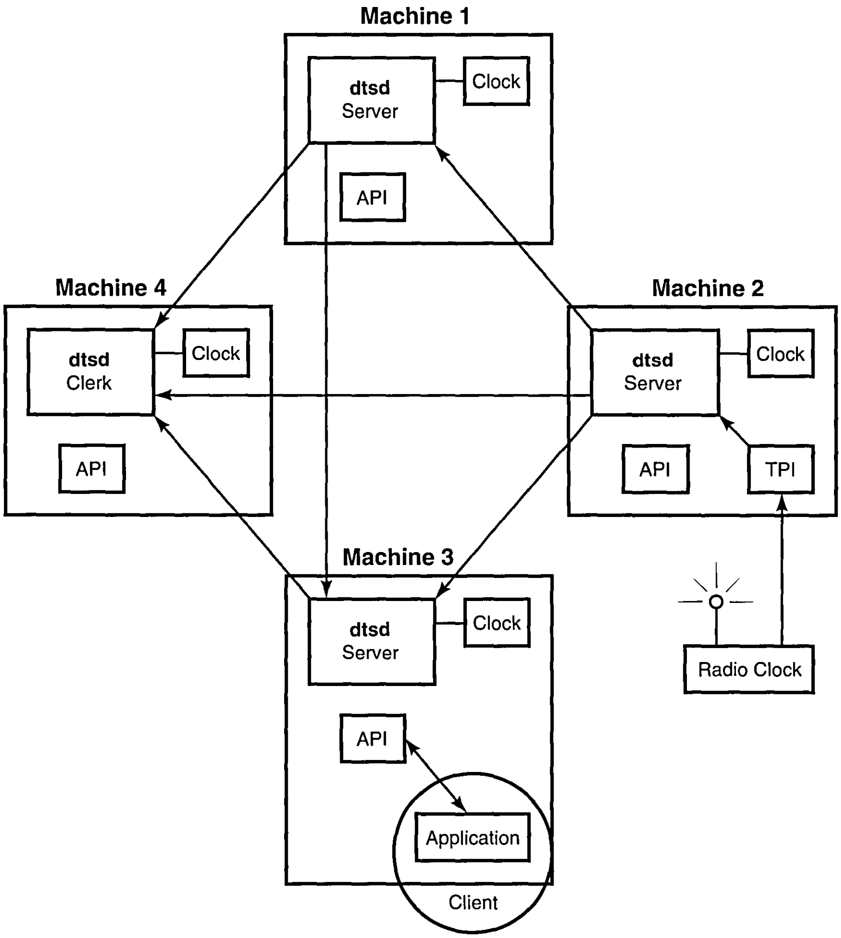
The *OSF DCE Administration Guide—Core Components* provides a detailed discussion of CDS and the DCE Directory Service, as well as detailing how the GDA works. The *OSF DCE Administration Guide—Extended Services* provides details about features of GDS and how it works.

2.1.3 The DCE Distributed Time Service

The DCE Distributed Time Service (DTS) synchronizes the clocks in network computer systems. DTS checks time synchronization and adjusts the clocks when the clock error exceeds a certain acceptable range, which you can set. Through its Application Programming Interface (API), DTS also provides timestamp manipulation functions to client applications.

Figure 2-5 shows the DTS relationships.

Figure 2–5. DTS Relationships



A DTS daemon (**dtsd**) runs on each DCE machine. Most of the **dtsd** daemons are configured as clerks. DTS clerks are responsible for receiving time values and adjusting the system clock accordingly. Some **dtsd** daemons are configured as servers. DTS Servers are responsible for synchronizing times among each other, as well as performing DTS clerk tasks.

DTS has a Time Provider Interface (TPI) that allows a server to import time values from outside time sources, such as radio, telephone, or satellite.

You use the DTS control program (**dtscp**) to perform DTS configuration and management tasks. However, the DTS synchronization functions run as background processes, and once DTS is installed and configured, the service does not require much intervention for system administration. You may need to adjust DTS configurations to meet varying conditions within your DCE cell. DTS administration tasks include the following:

- Registering DTS Servers as objects in the CDS namespace
- Configuring additional DTS Servers
- Setting the inaccuracy limit that forces synchronization, thereby bringing the inaccuracy back to an acceptable level

The *OSF DCE Administration Guide—Core Components* explains how DTS works and includes discussions of DTS time representation and basic time and clock concepts.

2.1.4 The DCE Security Service

The DCE Security Service enables clients and servers to prove their identities to each other. It offers integrity and privacy of communications and supports controlled access to resources. The DCE Security Service acts on behalf of principals. In DCE, principals are represented as entries in the DCE Security Service's database, the *registry*. These entries include users, servers, and computers.

The Security Service provides tools to help you administer Security on both the local machine and the cell. Tools for managing security on the local machine include the following:

- **passwd_export**

  This command updates the UNIX **/etc/passwd** and **/etc/group** files with current user information that is obtained from the database that is maintained by the DCE Security Service, the registry.

- **passwd_override**

  This file establishes overrides to the information that is contained in the registry.

Cell administration includes managing the Security Server (**secd**) and creating and maintaining information that is kept in the registry by using the **rgy_edit** command. The registry contains principals, groups, organizations, accounts, and administrative policies. Each cell has one registry database and replicas known as *slaves*.

The **rgy_edit** command allows you to set up accounts for foreign cells in your cell's registry, indicating that you trust the Authentication Service in the foreign cell to correctly authenticate its users.

Additional administrative tools enable you to back up and salvage the registry, create database replicas, and generate a new master key for encrypting passwords.

The DCE Security Service consists of several cooperating services and facilities. One of these services is the Registry Service described previously, which helps you manage user and group account information. In addition to the Registry Service, the DCE Security Service includes the following services and facilities that require very little or no system administration:

- The Authentication Service

  This service provides trustworthy identification of principals that are involved in network operations. A principal gains access to DCE by means of an account, which consists, in part, of the principal name and a secret key that the principal shares with the Authentication Service. A principal is indistinguishable from its account.

- The Privilege Service

  This service certifies a principal's identity and group membership. A principal's identity and group membership, also known as *privilege attributes*, determine a principal's access permissions to objects. The Privilege Service provides the privilege attributes that can be used to determine if a principal has the right to do what it wants to do.

- The DCE ACL Facility

  This facility determines a principal's access to an object by comparing entries in the object's ACL to the identity and group membership of the principal. The **acl_edit** command is the tool that users and administrators use to create, modify, and delete ACL entries. Other DCE components implement the ACL model provided by the DCE Security Service through their individual ACL Manager applications.

- The DCE Login Facility

  This facility initializes a user's DCE Security environment. This facility authenticates the user to the Security Service by means of the user's password, thereby establishing an authenticated network identity.

Refer to the *OSF DCE Administration Guide—Core Components* for detailed information about administration of the DCE Security Service. For additional information about the different parts of the DCE Security Service, refer to the *OSF DCE Application Development Guide*.

2.1.5 The DCE Distributed File Service

The DCE Distributed File Service (DFS) enables users to store and access data at remote locations. It extends the local file system model to the network, allowing users with proper authorization to have full or partial access to files that are exported by remote machines. To the user, it appears that all files are on the local machine.

The DCE Local File System (DCE LFS) component of DFS stores the file system data on the disk. In addition to supporting traditional UNIX file system features, DCE LFS is a log-based file system that enables quick recovery after a crash. DCE LFS supports filesets, which are hierarchical groupings of files that are managed as a single unit. By mounting a fileset at a location in the file tree, you make the fileset accessible to users. DCE LFS implements the ACL facility that is provided by the DCE Security Service to manage file access.

A UNIX file system that resides on a DFS File Server machine can also be mounted into the DFS file tree and accessed by DCE users. The UNIX file system is treated as a DFS fileset, but not all fileset operations are supported because DCE LFS functionality is a superset of the functionality that is provided by standard UNIX file systems.

The Cache Manager is the portion of a DFS client machine's kernel that communicates with DFS server processes by translating local file requests into remote procedure calls, if needed. The Cache Manager stores the requested files in a local disk or memory cache, where the files are available to users on that machine. For systems without a resident disk, the Cache Manager can run using a memory cache. You can customize certain features of the Cache Manager configuration, such as the cache size.

The File Exporter is the primary server process in DFS. The File Exporter resides in the kernel. This process handles requests from clients for files that it stores and manages.

DFS contains numerous components that handle administrative functions. DFS server processes manage the following tasks:

- Backing up DFS filesets that reside on server machines; this task is managed by the Backup Server
- Replicating filesets and ensuring consistency among copies; this task is handled by the Replication Server
- Storing information about the location of system and user files; this task is handled by the Fileset Location Server
- Ensuring that all file server machines in a cell have the same version of common configuration files and DFS binaries; this task is managed by the Update Server
- Monitoring server processes and restarting failed processes automatically; this task is accomplished by using the Basic OverSeer Server, or BOS Server

DFS provides several sets of commands that address administrative tasks. These commands include the following:

- The **fts** commands, which help you manage system and user filesets
- The **bak** commands, which help you copy filesets from the system to a backup tape and back to the system
- The **cm** commands, which help you reconfigure certain features of the Cache Manager and examine features of DFS
- The **bos** commands, which help you monitor, stop, and start processes, as well as perform some security tasks
- The **scout** command, which is a monitoring tool that collects and displays information such as disk usage

The DFS client and DFS Servers are described more fully in Chapter 4 and in the *OSF DCE Administration Guide—Extended Services*.

2.1.6 The DCE Diskless Support Service

The DCE Diskless Support Service provides the following features that can be integrated with operating systems of diskless workstation implementations:

- The Boot Service is used to obtain the boot image of a diskless workstation that is stored on a server. To use the Boot Service, you need to set up daemons and a configuration file.
- The Swap Service is used to access and administer the swap or paging space of a diskless workstation that is stored on a server. The Swap Service requires you to set up the appropriate daemon and swap server commands.
- The Diskless Configuration Service is used to configure diskless workstations. You need to enter and maintain the definition and values of diskless configuration data in a configuration file to set up this service.
- DFS operates as usual in the absence of a local disk by using a memory cache.

The DCE Diskless Support Service does not provide an operating system that is integrated with the preceding services to fully support diskless workstation operations. The operating system requires changes in order to use the tools that are provided by DCE to support a diskless workstation.

The DCE Diskless Support Service is described more fully in the *OSF DCE Administration Guide—Extended Services*.

2.2 How the DCE Components Work Together

Although DCE consists of distinct components, these components are integrated and interrelated. The following subsections summarize the relationships between components that have system administration functions.

Most DCE components rely on RPC, the DCE Directory Service, DTS, and the DCE Security Service. The interaction is often reciprocal. For example, RPC uses the DCE Security Service's Authentication Service to get keys, and the Privilege Service to securely associate clients and servers. The DCE Security Service, in turn, uses RPC for its communications.

The CDS component of the DCE Directory Service, DTS, and the DCE Security Service, along with RPC and DCE Threads, are the components that every DCE cell requires.

The GDA and GDS components of the DCE Directory Service and DFS are not required for a minimum DCE configuration. If these services are part of your cell, they rely on some or all of the services that are mentioned in the previous paragraph.

2.2.1 DCE Remote Procedure Call

An RPC server can store information about itself in CDS. An RPC client can look up location information about RPC servers in CDS when it wants to make a call to a particular server. CDS returns information that RPC libraries interpret as binding information and turn into a binding handle. The *binding handle* identifies the RPC server so that the RPC client can make its RPC call.

RPC uses the DCE Security Service to implement authenticated RPC, the process by which RPC clients and servers are identified to one another, and by which privacy and integrity of communications are maintained. To use authenticated RPC, clients and servers must be running as principals, have accounts, and have performed login operations.

Each RPC program is likely to require some administration of CDS namespace entries and directories, usually using **rpccp**, as well as some server-specific file administration.

2.2.2 The DCE Directory Service

CDS Servers and CDS clients use RPC and the DCE Security Service's Authentication Service to communicate with each other. CDS can also store information about the location of the RPC servers and interfaces that the RPC servers support.

CDS implements the ACL model that is provided by the DCE Security Service to ensure authorized access to directory data in CDS. Administration of ACLs is therefore required. To authenticate CDS interactions, CDS uses secure RPC that is provided through the DCE Security Service.

When you create entries in the CDS namespace, a timestamp accompanies the entry. The timestamp is used for propagation to replicas and the expiration of temporary entries. CDS relies on DTS to maintain clock synchronization in the network so that the timestamps are accurate.

CDS uses GDS to find names that are outside of the local cell by means of the GDA. Other DCE components interact with CDS for directory service (global and local), but only CDS and the application programs access GDS directly.

Unlike CDS and the other DCE components, GDS does not use RPC for its communications. GDS has its own security implementation and does not depend on the DCE Security Service. GDS conforms to the international standard X.500 protocols.

2.2.3 The DCE Distributed Time Service

Like CDS, DTS uses RPC to handle communications between DTS Servers and DTS clerks.

DTS registers the servers that synchronize system clocks in the network with CDS and also uses CDS to find DTS Servers and their locations.

To authenticate DTS interactions, DTS uses secure RPC that is provided through the DCE Security Service. DTS also uses DCE ACLs to control which users can execute the **set** or **show** options of **dtscp** commands.

2.2.4 The DCE Security Service

The DCE Security Service uses RPC for its communications. The DCE Security Service registers the location of its Security Servers (**secd** daemons) with CDS. Other servers in the network can use CDS to locate the Security Servers. These namespace entries must be managed by the administrator by using **cdscp**.

The DCE Security Service relies on DTS to maintain synchronized clocks so that passwords and *tickets*, which are used for obtaining network services, are properly timestamped and their expiration is enforced.

The DCE Security Service provides an ACL model for controlling access to objects that are managed by the DCE services. Based on this ACL model, objects and the ACLs on objects are controlled and managed by the DCE service that owns the object. The **acl_edit** command is used to manage access to principals, groups, and organizations that are registered in the cell namespace.

2.2.5 The DCE Distributed File Service

DFS uses RPC to handle requests for files at remote locations and to handle the communications between its clients and servers.

DFS registers its Fileset Location Server with CDS. When a user requests a file in the local cell or a foreign cell, DFS uses the DCE Directory Service to learn how to contact the Fileset Location Server that has information about the specified file. DFS also uses CDS to locate servers for administrative purposes.

DFS relies on DTS to maintain clock synchronization in the network.

DFS implements the DCE Security Service ACL model to determine access to files and directories and to manage the CDS namespace entries that are used by DFS. To authenticate DFS interactions, DFS uses secure RPC that is provided through the DCE Security Service.

2.2.6 The DCE Diskless Support Service

Like other DCE components, the Swap Service subcomponent of the DCE Diskless Support Service uses RPC to handle communications between the swap server and the diskless client.

The diskless client calls DFS with file system and cache information. DFS establishes the client's root file system and sets up the file cache in the client's memory.

Although the DCE Diskless Support Service is not integrated with the DCE Security Service, it can be if hardware support for secure booting is added by system vendors.

2.3 DCE System Administration Tasks

Like system administration in other distributed systems, DCE system administration can be divided into several areas. You perform some administration tasks only once, while others are part of your daily routine.

The system administration tasks that get you started and enable you to begin using DCE are planning, installing, building, configuring, and starting up DCE.

The *OSF DCE Release Notes* and the *OSF DCE Porting and Testing Guide* provide information about installing the DCE source tape and building DCE.

This guide contains planning information to assist you in installing executable files, configuring DCE, and starting up DCE.

Ongoing tasks, or maintenance tasks, consist of reconfiguring parts of DCE, monitoring DCE components, and performing routine management. These tasks are summarized in Chapter 6, and are described in detail in the *OSF DCE Administration Guide—Core Components* and the *OSF DCE Administration Guide—Extended Services*.

Chapter 3

Global and Cell Considerations

The purpose of Chapters 3 through 6 is to assist you in planning for the installation, configuration, and maintenance of DCE. For detailed information about installing the DCE source tape and building DCE, refer to the *OSF DCE Release Notes* and the *OSF DCE Porting and Testing Guide*. Part 2 of this guide describes the configuration process, including installing executable files, setting up a DCE cell, and configuring servers and clients.

This chapter discusses how to establish a DCE cell name. This chapter also describes how the DCE cell namespace is organized and provides guidelines for maintaining security and replicating parts of the cell namespace. The last portion of this chapter discusses what you need to consider as you plan for including DFS in your cell.

You need to answer a number of questions when planning for a distributed system. Keep in mind the following global considerations as you plan for DCE:

- How much do you think your environment will grow in the next few years? Do you anticipate rapid or relatively slow expansion of your network?

  If you think your environment will grow rapidly, consider setting up several cells representing smaller units of your organization. You can manage these smaller units as your network expands. As explained previously, members of each cell share a common purpose, and the cell is a unit of administration and security. If you anticipate slow expansion of your network, you may be able to establish one or more cells based on the organization that exists now. Consider how many administrators you will need to maintain your DCE cell, based on anticipated future growth.

- How much information does your environment have that needs to be distributed? How much do the users in your network share information?

  If there is a large volume of information that needs to be shared within your network, consider the amount of disk space that you require and the number of DFS File Server machines that you need.

- How much information updating do you require? Do the users in your network mainly look up information, or do they create and change information at their workstations?

  If information changes frequently and users in your network depend on the accuracy of that information, you need to consider how much you rely on replication. It is better to go to a central source of information for data that changes frequently. If users look up information but do not need to change the information that is shared with other users, you can rely more on replicated data.

- Is the most important data the most available data? Have you made plans to replicate this data?

  CDS, GDS, the Security Service, and DFS maintain master copies of their respective databases. Each CDS directory can be replicated separately. In addition to DFS databases, individual DFS filesets or groups of filesets can be replicated. GDS replication, also known as shadowing, can be done for a single object or an object and its subordinates (a subtree). The Security Service replicates the entire registry database. Because other components depend on the information that is managed by the Security Service and parts of the CDS namespace, that data needs to be available at all times. For example, the special character string **/.:** (the cell root) is stored in CDS and must always be available.

  Keep in mind that while replicating data improves availability, there is a cost in terms of performance and the amount of administration that is required.

- If your network has a gateway, are the servers located on the same side of the gateway as the clients that rely on those servers?

CDS Servers broadcast messages at regular intervals to advertise their existence to CDS clerks in the network. Clerks learn about servers by listening for these advertisements. Placing the servers and the clients that rely on them on the same side of the gateway facilitates efficient updates of information and a quick response to client requests. Additional administration is required if you rely on servers that are not available through the advertisement protocol, which is effective only in a local area network.

Consider how fast and how expensive links are if you are administering a cell that includes users in different geographic locations. You may want to keep more information locally to reduce your dependence on transmitting information across links.

- Is communication limited to your own cell, or do you need to communicate with other cells?

  For your cell to communicate with other cells, you must

  — Establish a unique GDS or DNS global name for your cell

  — Define your cell in one of the global directory services

  — Have at least one GDA in your cell

  You can set up a special account in your cell's Security registry for a foreign cell, indicating that your cell trusts the Authentication Service of the other cell, and a special account in the foreign cell's Security registry to represent your cell. (See the *OSF DCE Administration Guide—Core Components* for information about setting up these special accounts.) Even if you do not need to communicate with other cells now, consider whether you will need to communicate with other cells in the future. Be sure to establish a cell name with these future requirements in mind.

Your answers to these questions determine the basic requirements of your user environment. Use these requirements to help you decide on the optimum use of the DCE functions described in this and the following chapters.

3.1 Establishing a Cell Name

Before you can configure your DCE cell, you need to establish a cell name. The following subsections describe DCE naming syntax, naming conventions, and the procedure for obtaining a cell name.

3.1.1 Global Names

All DCE objects, including applications, machines, and users, have a global name. A global name is meaningful and usable from anywhere in the DCE environment. In DCE, global names begin with the special character string **/...**, which indicates the global root directory. Global names follow the syntax that is established by GDS or DNS.

3.1.1.1 GDS Global Names

GDS names are governed by a set of rules called a *schema*. Any valid X.500 name, including names that are provided by other standards supported by X.500, can be used as a cell name. An example of a schema that specifies a global, or absolute, name follows:

- Country, represented by **C=**
- Organization, represented by **O=**
- Organization Unit, represented by **OU=**
- Common Name (the user's name), represented by **CN=**

The hierarchy that is formed by the objects is called a Directory Information Tree (DIT). The following is an example of a fully qualified GDS global name, which is known as a Distinguished Name (DN):

/C=US/O=ABC/OU=Seattle/CN=Smith

Each part of a GDS name is composed of typed entries. Each entry is called a Relative Distinguished Name (RDN).

DCE uses an entry of a GDS name (such as **O=** or **OU=**) to store the attributes of cell objects. Following is an example of a DCE global name that uses the GDS format. Note that the string **/...** is prepended to the GDS name.

/.../C=US/O=ABC/OU=Seattle/sec/principal/smith

/.../C=US/O=ABC/OU=Seattle is the cell name, followed by the local cell portion of the name. In this example, the DCE cell attributes are stored in the Organization Unit portion of the GDS name.

3.1.1.2 DNS Global Names

DCE also supports global directory operations through the use of DNS. Following is an example of a global name that uses the DNS format:

/.../seattle.abc.com/sec/principal/smith

In DNS format, **/.../seattle.abc.com** is the cell name, followed by the local cell portion of the name.

3.1.2 Cell-Relative Names

In the two previous examples, **sec/principal/smith** is that part of the global name that resides in the local cell. The **sec/principal/smith** part of the global name can be used to construct a cell-relative name. Cell-relative names, also known as local names, are meaningful only from within the cell where the name entry exists. Cell-relative names begin with the special character string **/.:**, which replaces the global part of the name (the cell name). If you are in the **seattle.abc.com** cell, the following cell-relative name translates to the same global name shown in the previous examples:

/.:/sec/principal/smith

When you are entering a CDS name from the cell where that object is registered, you can use the cell-relative name. However, if you are entering a CDS name from another cell, you must use the global name, beginning with the character string **/...** (the global root).

CDS, GDS, and DNS naming conventions are described in more detail in the *OSF DCE Administration Guide—Core Components.* GDS naming conventions are described in the *OSF DCE Administration Guide—Extended Services.*

3.1.3 Choosing a DCE Cell Name

Choosing an appropriate DCE cell name is important for the following reasons:

- DCE cells that will participate in the global namespace must have unique names to differentiate them from cells in other organizations.
- A uniquely identified cell name is critical to the operation of DCE Security; this name is the basis for authentication in your cell.
- Both GDS and DNS expect global cell names to have a certain format. Choose a name that conforms to either GDS or DNS naming conventions.
- DCE does not currently support cells that are registered simultaneously in GDS and DNS.
- *There is no supported way to change the name of a cell.* Choose your cell name carefully.

Note that cell names are case insensitive; that is, the name **MyCell** is equivalent to the name **MYCELL.** Note also that when comparing cell names, DCE routines change the names to all lowercase characters before making the comparison. Cell names *must not* contain an @ (at sign).

The following subsections describe conventions that apply to GDS and DNS names.

3.1.3.1 DCE Cell Name Conventions for GDS

A fixed set of two-letter codes must be used to indicate the Country (**C=**) attribute of a GDS global name. In addition, a country may have standard Organizations (**O=**) where your organization is registered as a code. Your organization may also have conventions that apply to the way an Organization Unit (**OU=**) is represented. Check with your naming authority for the exact conventions (see Section 3.1.4.1).

Refer to the *OSF DCE Administration Guide—Extended Services* for details about naming rules, including valid characters, restrictions, and maximum name sizes for GDS names.

3.1.3.2 DCE Cell Name Conventions for DNS

If you plan to use DNS as your global directory service, either immediately or in the future, your DCE cell name must follow the ARPA Internet Domain System conventions for site names. If you are already an Internet site, you can create one or more cells subordinate to your Internet domain name, depending on how your site is organized.

The following conventions govern an Internet-style name:

- The name needs to have at least two levels; for example, **abc.com** or **sctech.edu**. The names in the first two levels are registered with the Network Information Center (NIC), which is the naming authority for DNS names. Registration request information is detailed in Section 3.1.4.2.

- Although there is no restriction on the length of a name, a short name is more convenient to type.

- The name can contain any number of fields in addition to the two required levels, which are conventionally separated by periods.

- The name needs to end in a suffix that indicates a kind of institution. This last field is the most significant one, in contrast to a GDS name, which begins with the most significant field. The standard suffixes are as follows:

  — **.com** for businesses and other commercial organizations

- — **.org** for noncommercial organizations
- — **.edu** for educational institutions
- — **.gov** for government institutions
- — **.mil** for military institutions
- — **.net** for network support organizations
- — *.xx* for two-letter country codes (such as **.de** for Germany and **.fr** for France) that conform to the International Organization for Standardization (ISO)

Refer to the *OSF DCE Administration Guide—Core Components* for further information about naming rules, including valid characters, restrictions, metacharacters, and maximum name sizes for CDS and DNS names.

3.1.4 Obtaining a DCE Cell Name

If you plan to create a private cell and do not ever intend for it to communicate with cells that are outside your organization, you are not required to obtain a globally unique cell name. However, in order for your cell to communicate with other cells that are outside your organization, you need to have a GDA running, and before you configure your cell you need to obtain a globally unique cell name from the GDS or DNS global naming authorities. The name can be one that already exists and is in use, or you can create a new one. This registration must be completed before you begin to configure the cell namespace. It is recommended that you obtain a unique global name for your cell even if you do not initially use a global directory service to communicate with other cells, so that you can do so in the future.

Keep in mind that once you have named your cell, its name cannot be changed. Choose your cell name carefully.

3.1.4.1 Obtaining a Unique GDS Cell Name

To obtain a unique GDS cell name, contact the administrator who is in charge of the portion of the DIT under which you want to name your cell.

For example, in the United States, the American National Standards Institute (ANSI) delegates X.500 names that are subordinate to the entry **/C=US**. Suppose you are an employee of ABC, a U.S. corporation that is interested in participating in a worldwide X.500 directory. If you wanted to configure a single cell whose name is **/C=US/O=ABC**, you would contact ANSI to reserve ABC as a unique organization name. Similarly, if you wanted to configure multiple cells in your organization and name them based on organization units, you would contact a naming authority within your company to establish a cell entry, such as **/C=US/O=ABC/OU=Sales**.

Send X.500 name registration requests to

American National Standards Institute
11 West 42nd Street
New York, NY 10036

Telephone Number: (212) 642-4976

It is the responsibility of the person making the request to ANSI to be sure that your organization does not send more than one request for an organization name. Once you receive your organization name, it is recommended that your organization set up a central administrative authority to manage names that are subordinate to the organization name.

3.1.4.2 Obtaining a Unique DNS Cell Name

To obtain a unique DNS name, contact the administrator who is in charge of the subtree under which you want to name your cell. Send registration requests to the NIC at the following Internet address, telephone number, FAX number, or mailing address:

HOSTMASTER@NIC.DDN.MIL

Telephone Number: (800) 365-3642
between the hours of 7:00 a.m. and 7:00 p.m. Eastern Standard Time

FAX (703) 802-8376

Government Systems, Inc.
Attention: Network Information Center (NIC)
14200 Park Meadow Drive
Suite 200
Chantilly, VA 22021

3.1.5 Defining a Cell in GDS or DNS

After you obtain a unique name for a cell, configure the cell, and initialize the cell namespace, the next step in establishing the intercell directory service is to create an entry for the cell in one of the global namespaces. You can use the **cdscp** subcommand **show cell** to obtain the data that you need to create or modify a cell entry in DNS or GDS. The *OSF DCE Administration Guide—Core Components* contains an example of the output of the **show cell** command. The data you obtain from the command is what CDS uses to contact servers in foreign cells.

GDS stores cell information in the **CDS-Cell** and **CDS-Replica** attributes. You must add these two attributes to an existing GDS entry for that entry to become a cell entry. (See the *OSF DCE Administration Guide—Extended Services* for more information.)

To create a cell entry in DNS, an administrator must edit a data file that contains *resource records*. The *OSF DCE Administration Guide—Core Components* provides detailed information on how to create a cell entry in DNS. The *OSF DCE Administration Guide—Core Components* also includes information on the following topics: how the GDA works and how to manage the GDA, defining a cell in DNS, and defining a cell in GDS.

3.2 The Cell Namespace

An integral part of planning for a DCE cell is understanding the organization of your cell namespace. Consider the following as you plan the organization of a cell in your network:

- Are the security requirements maintained?
- Does the organization of the cell facilitate network traffic where the data sharing needs are the greatest?
- How will you manage the administrative accounts that will be created for each DCE service during the configuration process?
- What are your DFS administrative domains; that is, the groups of DFS Servers that are administered as a unit? Can you group the servers for more efficient administration?

3.2.1 Determining Cell Boundaries

In DCE, the boundaries of a cell are equivalent to the boundaries of the cell namespace. A small organization can consist of one cell. A large organization can have many cells. The primary factors in determining a cell's boundaries are the common purpose and trust that are shared by the cell's principals. Principals within a cell can belong to groups that share the same privileges. Members of a group share the same level of trust and are authorized to perform certain actions.

Because there is a set of administrative tasks that are associated with setting up and maintaining each cell, it is reasonable to keep the number of cells in your organization to a minimum. However, the level of trust that is shared by groups of principals is a more important consideration than administrative overhead.

3.2.2 Keeping Cells Stable

Once you decide how many cells you need and where the boundaries of those cells will be, make an effort to keep your cell structure stable. Servers are not easily moved from one cell to another, so be sure to plan your namespace structure carefully in order to minimize reconfiguration. If you do need to move a host from one cell to another, you must do the following:

- Delete or move the server processes from the host.
- Delete or move any databases.
- Change the name of the cell in the host's configuration files (**dce_cf.db** and **pe_site**).
- Reconfigure the host in the new cell.
- Delete any namespace or registry entries for the host in the old cell.

3.2.3 Types of Cell Namespace Entries

The following subsections describe the different types of entries that comprise the cell namespace. These entries are created when you follow the default configuration path that is described in Part 2. The *OSF DCE Administration Guide—Core Components* and the *OSF DCE Administration Guide—Extended Services* provide details about the names that the DCE components use. The cell namespace can be divided into three major parts:

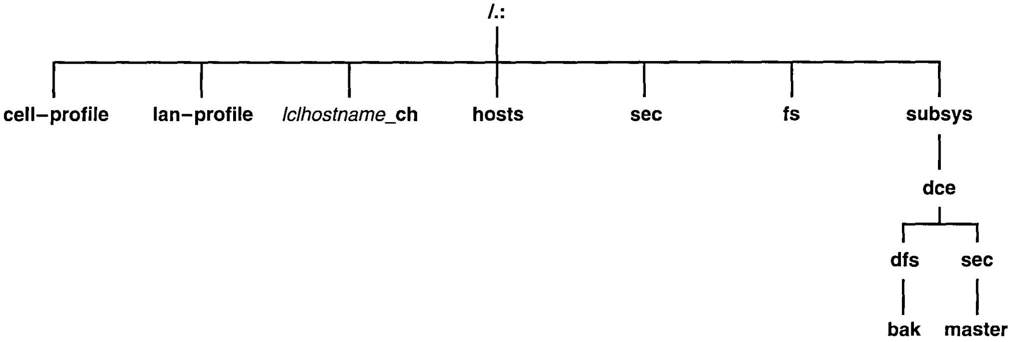
- The CDS part of the namespace
- The Security part of the namespace
- The DFS part of the namespace (the filespace)

Each of the DCE services maintains its own namespace within the DCE cell namespace. DFS maintains its own namespace to ensure consistency among many files. The Security Service maintains its own namespace to ensure that the DCE cell remains secure. Clients of these two services query CDS for binding information that enables them to find Security or DFS Servers. The points where the binding information is stored serve as mount points in the CDS namespace for the namespaces that DFS and the Security Service manage. This transition point between two namespaces is called a *junction*.

The **/.:/sec** directory is the junction from the CDS part to the Security part of the cell namespace, and the **/.:/fs** directory is the junction from the CDS part to the DFS part of the cell namespace.

Figure 3-1 shows the top level of the cell namespace. In some cases, the names in the cell namespace are fixed (or well known) and cannot be changed. In other cases, you can choose a different name from the one that is listed. (See the *OSF DCE Administration Guide—Core Components* for more information about which names are well known.) In Figure 3-1, **/.:** and **cell-profile** are well-known names.

Figure 3–1. Top Level of the Cell Namespace



You can use the **cdsbrowser** or **cdscp** utility to view the CDS namespace, including the **sec** and **fs** junctions. You can use commands such as **ls** to see the contents of the DFS part of the cell namespace and the **rgy_edit** command to see the contents of the Security portion.

3.2.3.1 CDS Namespace Entries

The CDS namespace contains entries for servers, hosts, CDS clearinghouses (collections of directory replicas stored at a particular server), RPC profiles, RPC groups, and subsystems. The entries have a CDS type of Directory or Object, indicating the kind of CDS object to which the name refers. A CDS directory is a container in which objects are stored. CDS uses directories to organize groups of object entries.

In addition, the CDS namespace provides specialized services for other DCE components, such as location information that is contained in the Fileset Location Database (FLDB), which is the database that maps filesets to the File Server machines on which they reside.

Profiles that are cataloged in the CDS namespace specify a search path through the Directory Service. The cell profile (**/.:/cell-profile**) stores the location of the servers that are available in the cell, regardless of the physical location. In a geographically dispersed cell, servers can be located in different cities or even different countries. The LAN profile defines alternate servers that can be used in situations where geographic proximity is important. For example, **/.:/lan-profile** is the default LAN profile that is used by DTS. This profile contains entries for the DTS Server local set. If a cell spans more than one LAN, another layer can be created below **/.:/lan-profile** to specify the location of the profile for each part of the cell. For example, in a cell that encompasses two LANs, you can direct hosts on one LAN to **lanA-profile** and hosts on the other LAN to **lanB-profile**. For information on setting up multiple LAN profiles, see Chapter 8.

3.2.3.2 Security Namespace Entries

The types of Security entries are as follows:

- **principal**

  This type of entry contains an individual principal.

- **principal** directory

  This type of entry contains individual principals or one or more principal directories, or both.

- **group**

  This type of entry contains an individual group.

- **group** directory

  This type of entry contains individual groups or one or more group directories, or both.

- **org**

  This type of entry contains an individual organization.

- **org** directory

  This type of entry contains individual organizations or one or more organization directories, or both.

- **policy**

  This type of entry contains Security policy.

When you (or an application) are accessing an entry in the Security part of the namespace, the name of the entry alone provides enough information for the Security Service to work with. For example, the Security Server knows that the login name is a principal name that is registered in the Security part of the namespace; */.:/principal_name*, */.../cell_name/principal_name*, and *principal_name* are all valid ways of representing the name that you use to log in.

When you use the **rgy_edit** command, you specify the type of object you will operate on before you operate on it. For example, to change account information that is associated with the principal **smith**, you specify that you want to operate on a principal, and you then enter the principal name **smith**. The **rgy_edit** command deals with the following three types of objects:

- Principals
- Groups
- Organizations

The *OSF DCE Administration Guide—Core Components* explains how to use the **rgy_edit** command to display information that is related to principals, groups, organizations, and accounts.

In contrast to **rgy_edit**, which operates on those objects that are registered in the Security space, the **acl_edit** command operates on all objects in the namespace. The **acl_edit** command requires the object's fully qualified pathname, as shown in the following example:

/.:/sec/principal/smith

and not simply the following:

smith

The following parts of the namespace comprise the Security namespace:

- **/.:/sec/principal**
- **/.:/sec/group**
- **/.:/sec/org**
- **/.:/sec/policy**

3.2.4 CDS Namespace Replication Considerations

Directory replication is the most reliable way to back up the information in your CDS namespace. Because the CDS data is replicated by directory, when you replicate a directory, all of the entries in it are automatically replicated. Use the **cdscp** control program to create replicas of directories at a CDS clearinghouse. Clearinghouses need to be created in the root directory (**/.:**) of the cell namespace.

Follow these guidelines for replicating parts of the cell namespace:

- The root directory (**/.:**) is automatically replicated when you create a clearinghouse.
- You should have at least two copies of each CDS directory to ensure that the entire namespace is available at all times. (See the *OSF DCE Administration Guide—Core Components* for further information about backing up CDS information.)

3.3 Planning for Access Control

When planning for access control, it is important to keep the level of access control in your cell restrictive enough to ensure that security is maintained. A special set of individuals or a special group can be given permission to create accounts and groups in the root directory of the Security space. A group called **acct-admin** is created when you configure DCE. The **acct-admin** group is the only group that can create accounts and groups in the root directory of the Security space.

While maintaining an adequate level of security in your cell, you also need to consider the requirements of administrators who are maintaining DCE services when you set access control levels. For example, if one person is responsible for administration of DFS in your cell, that person may need to add servers to the Security and CDS namespaces. On the other hand, an administrator who is responsible for the Security Service manages the Security Server but does not control the DFS filespace.

Following are some of the groups that are created when you configure DCE using the DCE configuration script:

- **sec-admin**

  This group administers Security Servers, registry replication, and other Security functions.

- **cds-admin**

  This group administers CDS Servers, CDS replication, and other CDS functions.

- **dts-admin**

  This group administers DTS Servers and related DTS functions.

- **dfs-admin**

  This group administers DFS File Servers and related DFS functions.

(See Section A.2.3 for a list of DCE groups created by the DCE configuration script.)

In addition to the administrative groups, individual users need permission to control some information that is kept in the registry database. For example, individual users need to be able to change their password, home directory, or login shell.

3.4 The Filespace

The following subsections contain guidelines for planning your cell's filespace. The *OSF DCE Administration Guide—Extended Services* explains some of these planning considerations in more detail.

The filespace begins under the cell root at the **/.:/fs** junction to DFS from the CDS namespace. The notation **/:** is set up by default to be equivalent to **/.:/fs**. Thus, the notation **/:/usr/***user_name* is equivalent to **/.:/fs/usr/***user_name*.

Some parts of DFS run in the host machine's kernel. This kernel function must be present on your machine before you run DFS.

3.4.1 DFS Administrative Domains

A *DFS administrative domain* is a collection of machines in the same cell that are configured for administration as a single unit. In a single cell you can have one or many administrative domains, depending on the size of your organization. Organizing DFS Server machines into different administrative domains simplifies the management of the cell filespace by creating smaller units for administration. All machines within an administrative domain must be in the same cell.

3.4.2 DFS Administrative Lists

DFS administrative lists are files that define the principals and groups that can perform actions affecting specific server processes on a server machine. There is one DFS administrative list for each DFS server process that is running on a machine. For example, a server's **admin.bos** file defines who has administrative rights to the BOS Server (**bosserver**), and thus determines who can manipulate and maintain server processes on that one server. Groups, as well as individual users, can be placed on an administrative list. Each server machine stores administrative lists for its processes on its local disk. A process automatically creates its initial administrative list when it is started, if the list does not already exist on the local disk of the machine.

3.4.3 Determining the Roles of DFS Machines

Follow the recommendations in the the *OSF DCE Administration Guide—Extended Services* when you assign roles to the DFS machines in your cell.

The first DFS machine that you configure during DCE installation and configuration (described in Chapter 9) needs to function as a System Control Machine. The System Control Machine is the server that distributes DFS configuration information. Next, you configure a Fileset Location Database Server, which is the server that maintains the fileset location database. The DCE installation and configuration script assumes that the **root.dfs** fileset, which is the fileset that corresponds to the top (**/.:/fs**) level of the file tree, is located on the Fileset Location Database Server. (See Section 3.4.5 and the *OSF DCE Administration Guide—Extended Services* for further information about **root.dfs.**)

Machines that you configure as DFS Servers can run the processes that are required to be File Servers. Be sure the machine you choose has enough space to store DCE LFS filesets. The amount of free space you need depends on how much data you plan to store in DCE LFS filesets. Filesets on File Servers can store DFS client binaries in addition to user files. These filesets can also be distributed on other File Server machines in your cell. In addition, if your domain has only one server machine, this machine must run all processes and fill all required machine roles. For example, in addition to being a System Control Machine, this machine must be a File Server and a Fileset Database Server. If your domain has three or more DFS Server machines, three machines need to store DFS databases. An odd number of DFS database machines is recommended.

3.4.4 Setting Up the DFS File Tree

Follow the recommended conventions in this section when you set up your DFS file tree. (See the *OSF DCE Administration Guide—Extended Services* for more information about this process.)

Below **/.:/fs** are directories that help organize your DFS environment, such as

- The **common** directory

  This directory contains programs and files that are needed by users who are working on machines of all system types, such as text editors or online documentation files. The **common/etc** directory is a logical place to keep the central update sources for files that are used on all DFS client machines.

- The **public** directory

  This directory contains files that users want to make available to everyone, including foreign and unauthenticated users.

- The **sys_type** directory

  This directory contains binaries for each system type that you use as a File Server or client machine. If you plan to use the **@sys** variable in pathnames, you need to use standard names to represent system types.

- The **usr** directory

  This directory contains the home directory of each DFS user in a cell and any foreign users that are granted a local account. Users and system administrators can protect this directory so that only locally authorized users can access it. If your cell is quite large, you can divide user home directories in multiple directory listings to facilitate quicker directory lookups.

- The **src** directory

  This directory contains source filesets, such as those for DFS source files.

3.4.5 Setting Up Filesets

Consider the following recommendations and restrictions when you set up filesets:

- Fileset names must be limited to 102 characters or less.

- Every cell must include **root.dfs**. The root fileset can be a DCE LFS fileset or it can be a non-LFS fileset (a non-DCE LFS file system). If

root.dfs is a DCE LFS fileset and you plan to use replication, you need to follow the steps described in the *OSF DCE Administration Guide—Extended Services*, which describes how to create **root.dfs** as a DFS LFS fileset and create a read/write mount point for the fileset below the top level of the cell's filespace.

- You should use a common prefix when naming related filesets. This aids in manipulating and grouping related filesets. It also relates the fileset's name to its mount point.
- You can group filesets on the same partition of a File Server machine. This can localize the effects of an outage, but you also need to consider factors, such as the number of File Server machines and load balancing, before grouping filesets.
- You can replicate filesets for load balancing and to make fileset contents more available. Replication is appropriate for filesets that are read much more often than they are written, such as filesets containing installed executable files. Replication is not supported for non-LFS filesets.
- Consider the disk space a fileset requires before setting up filesets.
- If a domain includes DCE-based diskless machines, you need to create the fileset that serves as the potential top-level directory of a diskless machine. (See the *OSF DCE Administration Guide—Extended Services* for details.)

3.4.6 Using @sys and @host Variables

Follow the suggested conventions in the *OSF DCE Administration Guide—Extended Services* when using the **@sys** and **@host** variables in certain pathnames. When the Cache Manager encounters one of these variables, it substitutes a string that consists of the local machine's architecture and operating system type for **@sys** or the hostname for **@host**, causing a certain directory to be used. Using **@sys** and **@host** is helpful when you are constructing symbolic links from the local disk to DFS. You can create identical symbolic links on all machines, but each machine transparently accesses the files that are appropriate to its system name or hostname. The **cm sysname** command sets and displays the current value for **@sys**.

Chapter 4

Client and Server Considerations

This chapter describes configurations for DCE Client machines, the different types of DCE server machines, DCE Administration machines, and DCE Application Development Environment machines. A DCE Client machine can run client code of every DCE service. DCE server machines are configured to run a certain set of software. This software is made up of at least one daemon and, in some cases, one or more additional programs that comprise the server side of a DCE component. DCE server machines also run the software that makes up the DCE Client configuration. DCE Administration machines contain the administration programs for the DCE servers, in addition to DCE Client software. The DCE Application Development Environment configuration contains files, such as header files, that are needed by DCE application programmers, in addition to the DCE Client software.

In addition to the space that is required for DCE programs listed in this chapter, you need to allow between 65 and 100 megabytes for paging space. For DCE's OSF/1 reference platform, allow approximately 2 megabytes for DFS libraries that must be linked into the UNIX kernel. For AIX, allow approximately 3 megabytes for the extensions that are dynamically loaded into the UNIX kernel.

(See the *OSF DCE Porting and Testing Guide* for a list of all the files that are provided for a DCE component.)

The sections of this chapter are presented in the order in which you need to approach planning for configuring DCE machines. Table 4-1 summarizes the configurations that are discussed in this chapter and the space requirements for each reference platform.

Table 4–1. Space Requirements for DCE Machines

	Approximate Size (bytes)	
Configuration	**OSF/1**	**AIX**
libdce	6,500,000	5,400,000
DCE Client	10,700,000	10,400,000
CDS Server (and GDA)	2,060,000	1,150,000
Security Server	1,420,000	1,070,000
DTS Server	0	0
GDS Server	8,550,000	4,540,000
Composite DFS Server	11,900,000	10,200,000
DCE Administration	9,600,000	5,850,000
DCE Application Development	2,020,000	1,790,000

The size requirement for the DTS Server is 0 (zero) because the software for both the DTS client and DTS Server is the same, and is already included in the DCE Client.

4.1 Determining Requirements for DCE Client Machines

The following subsections describe the planning considerations that are involved in setting up DCE Client machines. All DCE machines, including DCE server machines, are also DCE Clients.

4.1.1 Files Installed on DCE Client Machines

This section gives an overview of the software that is installed on DCE Client configurations. Additional details are provided in Sections 4.1.2 through 4.1.7.

Minimum DCE Client

A minimum DCE Client configuration contains client services for RPC, CDS, Security, and DTS. This configuration is currently not supported by the DCE configuration script.

Full DCE Client

In addition to the files that are needed for the minimum DCE Client configuration, the full DCE Client contains client side files for GDS and DFS. (See Table 4-1 for the approximate space requirements.) The files for a full DCE Client are as follows; this list includes the files that are needed for a minimum DCE Client:

- Daemons

  cdsadv, cdsclerk, dtsd, rpcd, sec_clientd

- Utilities

  acl_edit, cdscp, dce_login, dtscp, getcellname, kdestroy, kinit, klist, rgy_edit, rpccp, uuidgen

- Data Files and Shell Scripts

  cds_attributes, cds_globalnames, cdscp.bpt, cdscp.mbf, dce_config, dce.clean, dce.rm, dtscp.bpt, localtime, posixrules, rc.dce, rc.dfs

- Libraries

  libdce

- Message Catalog Files

  dcecds.cat, dcecfg.cat, dcedts.cat, dceevt.cat, dcekdb.cat, dcekdc.cat, dcekrb.cat, dcerpc.cat, dcesad.cat, dcesec.cat, dcethreads.cat, gdsditadm.cat, gdsproc.cat, gdssysadm.cat, idl.cat, uuidgen.cat

- DFS Message Catalog Files

  dfsasy.cat, dfsbak.cat, dfsbbs.cat, dfsbdb.cat, dfsbtc.cat, dfsbtm.cat, dfscmd.cat, dfscmp.cat, dfsdau.cat, dfsdcl.cat, dfsepi.cat, dfsfsh.cat,

dfsfts.cat, dfshst.cat, dfslgb.cat, dfssal.cat, dfstkm.cat, dfsubk.cat, dfsupd.cat, dfsvls.cat, dfsxcr.cat, dfsxvl.cat

- Files on AIX Only

  The following files are needed on the AIX platform only; they are kernel extensions. On the OSF/1 platform, they are already linked into the kernel.

  config_kern_ext, dtskernext, dtsloadobj, load_kern_ext, query_kern_ext, unload_kern_ext

- DFS Clients Only

  The following files are needed only if the DCE Client machine is also a DFS client machine:

  dfsbind, dfsd

  The following are optional on a DFS client:

  bos, cm, fts

- AIX DFS Clients Only

  cfgdfs, cfgexport, dfscmfx.ext, dfscore.ext, dfsloadobj, export.ext

The following subsections describe the executables that run on a DCE Client machine.

4.1.2 RPC Client Programs

A DCE Client contains the following RPC programs:

- The **rpcd** daemon must run on any machine that has an RPC server process that exports an interface with dynamic bindings. The **rpcd** daemon is used to register binding information. The **rpcd** daemon must run on every DCE machine because on every DCE machine there are client-side daemons that export interfaces. For example, the **sec_clientd** daemon exports the **override** interface and the **dtsd** daemon exports the **acl** interface. (The **sec_clientd** and **dtsd** daemons are discussed in Section 4.1.3 and Section 4.1.5, respectively.)

  The **rpcd** daemon must be running before you configure any other DCE services because DCE services need to register their endpoints with **rpcd**. Only one **rpcd** daemon is needed on a machine. In fact, only one

rpcd daemon can run on a machine at a time because **rpcd** uses a well-known port.

Network interfaces, routing services, and other network services must be available before RPC starts. The **rpcd** daemon is started in the **/etc/rc.dce** file. The **/etc/rc.dce** file can be invoked by other **rc** files, such as **/etc/rc** and **/etc/rc.local**, so that DCE services can be brought up each time a machine boots.

- The RPC control program (**rpccp**) is a utility that allows you to browse, update, add, and delete the RPC attributes of entries that are stored in the CDS namespace and the endpoints that are managed by local and remote **rpcd** daemons.

4.1.3 Security Service Client Programs

Every DCE Client machine runs the **sec_clientd** daemon. The **sec_clientd** daemon takes the place of the machine principal. Most principals are interactive users, but the machine principal is not. The **sec_clientd** daemon performs the processing that is necessary so that other daemon processes on the machine appear to be running with the machine's identity.

The **sec_clientd** daemon periodically refreshes the ticket-granting ticket for the machine's principal. A DCE Client machine must have a valid ticket-granting ticket in order for a principal to use DCE services. The **sec_clientd** daemon also exports the interface that assures a Security client that it is actually contacting the real Security Server when the client requests a ticket-granting ticket from the Security Server.

(See the *OSF DCE Administration Guide—Core Components* for more information about ticket-granting tickets.)

4.1.4 CDS Client Programs

The DCE Client runs the following CDS processes:

- The CDS advertiser, the **cdsadv** process, does the following:
  - — Allows applications to locate and communicate with **cdsd** servers

  - — Starts any needed CDS clerks (**cdsclerk**)
  - — Creates the cache that is shared by local CDS clerks

- The **cdsclerk** is an interface between CDS client applications and CDS Servers. A clerk must exist on every machine that runs a CDS client application. One **cdsclerk** process runs for each DCE principal on a machine that accesses CDS. The CDS clerk handles requests from client applications to a server and caches the results that are returned by the server. Because the results of the server request are cached, the clerk does not have to go repeatedly to the server for the same information. All CDS clerks on a machine share one cache. One clerk can serve many client applications.
- The CDS control program, **cdscp**, is a command interface that is used to control CDS Servers and clerks and to manage the namespace and its contents. (See the *OSF DCE Administration Guide—Core Components* for more information about the CDS control program.)

4.1.5 DTS Client Programs

The DCE Client runs the following DTS processes:

- The **dtsd** daemon is set to be a client or a server. On a client machine, **dtsd** synchronizes the local clock.
- The **dtscp** program allows you to administer DTS, including configuring the **dtsd** daemon as either a client or a server.

4.1.6 GDS Client Programs

This section describes the programs that make up the client side of GDS. However, the GDS client software is not installed by the DCE configuration script when a DCE Client is being installed. To install the GDS client software or any other GDS configuration, use the GDS server installation menu. Then, use **gdssysadm** to further configure GDS.

If GDS is installed, the DCE Client runs the DUA. The DUA, which is the client side of GDS, sends requests to the GDS Server process, the DSA. The DUA consists of the following processes:

- The **gdscache** process caches user data and stores data that is used for regulation purposes locally.
- The **gdscstub** process handles all outgoing requests to remote DSAs.
- The **gdscacheadm** program supports administration of the contents of the local DUA cache database.
- The **gdsipcchk** program verifies the IPC-state information that is contained in the shared memory area of a GDS installation.
- The **gdssysadm** program supports administration of the local GDS installation, such as configuring GDS, activating servers, and backing up the database.

A machine that is running only the client side of GDS can access GDS Servers on other machines, or one machine can run both the client and server portions of GDS. Machines that are running just the DUA are known as *client systems*. Client systems can access directory information on server machines without having to store that information.

4.1.7 DFS Client Programs

If DFS is installed, the DCE Client runs the following processes:

- The Cache Manager process, **dfsd**, initializes the Cache Manager in the kernel, alters the configuration settings, and starts the background daemons.

  The Cache Manager is responsible for the local caching of file and directory data on machines that are used as DFS clients. When the Cache Manager starts, it initializes the cache. When a client retrieves part of a file from a remote File Server, the Cache Manager keeps a copy of that part of the file on the client machine's local disk. As long as that part of the file does not change, the locally cached copy remains available to the client. A new copy is retrieved from the File Server machine only when another process changes the cached portion of the file. The Cache Manager also caches directory and fileset location information.

- The **dfsbind** process does the following:
  - Obtains cell location information from CDS
  - Responds to Security requests on behalf of the DFS kernel processes by making calls to the Security Server

4.2 Determining Requirements for DCE Server Machines

The following subsections describe the planning considerations that are involved in setting up DCE server machines.

4.2.1 Files Installed on DCE Server Machines

The following subsections list the files that must be installed on each of the different DCE server machines. Table 4-1 lists the approximate space requirements for each server machine. Note that because all DCE servers are also DCE Clients, the files that are described in Section 4.1 must also be installed on server machines. Therefore, add the appropriate server space requirements to the DCE Client machine space requirements to reach an approximate total space requirement for the configuration that you are planning.

4.2.2 RPC Server Programs

There are no RPC server programs other than the programs that run on the DCE Client.

4.2.3 Security Server Processes

Every cell has one master DCE Security Service machine and can also have slave DCE Security Service machines. The following processes run on a DCE Security Service master or slave server machine:

- The Security Server, or **secd** process, implements the Authentication Service, the Privilege Service, and the Registry Service.
- The **sec_create_db** program initializes the Security database. You give this command an option indicating whether you want to create a master or slave Security server on the machine.

Keep the following considerations in mind when you are planning for Security servers:

- The node that runs the master Security server must be highly available and physically secure. Consider placing the master Security server machine in a locked room and keeping a log to record who accesses the machine.
- Be sure to move the master Security server before removing the node from the network or shutting down the node for an extended period of time. Modifications are made to the master Security server and propagated to slaves throughout your cell. If the master Security server is unavailable, no updates can be made.
- A cell can have only one master Security server. If you plan to make one cell out of several existing cells with independent master Security servers, you must first merge their registries.
- If the host that contains the master Security server goes down, the hosts that have slave servers can still provide registry information, so consider having a number of slaves in your network. 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 slave Security servers you need to have.

(See the *OSF DCE Administration Guide—Core Components* for further information about planning for the DCE Security Service.)

4.2.4 CDS and GDA Server Processes

A CDS Server stores and maintains object names within a cell and handles requests to create, modify, and look up data. A CDS Server machine is by default configured as a GDA server as well, running the **gdad** daemon. There must be a GDA server running in a cell in order for the cell to communicate with other cells.

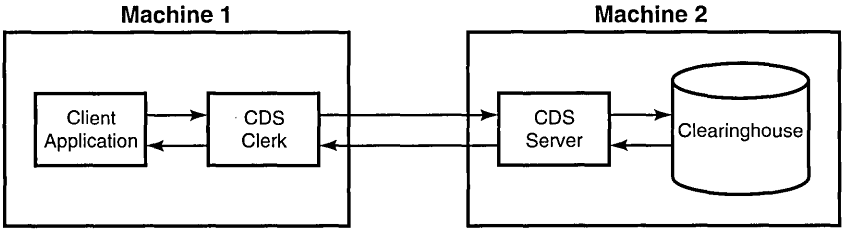
The following processes run on a CDS Server machine:

- The CDS daemon, **cdsd**, is the CDS Server process.
- The **cdsadv** on a DCE Client machine, receives server advertisements to find out what servers are available. On a CDS Server machine, it also sends server advertisements.

When preparing for CDS, you need to select server nodes that store and maintain the clearinghouses (CDS databases) in the cell.

Figure 4-1 shows a client application that sends a request to the CDS clerk, which in turn communicates with the CDS Server. The server performs a database lookup or update, depending on the request. The response is then sent back to the client application.

Figure 4–1. CDS Client and Server Machines



Keep the following guidelines in mind in order to achieve reliability, optimum performance, and data availability:

- Choose dependable nodes. A CDS Server wants to avoid downtime as much as possible and needs to be restarted quickly when downtime occurs. The CDS Server needs to be one of the first systems available on the network because client applications and other DCE servers rely on the CDS Server for up-to-date information. The CDS Server initializes the CDS namespace when you configure DCE.

- Use reliable network connections. This helps to ensure that all servers that are maintaining directory replicas can be reached when CDS performs a skulk. Skulks are periodic updates that check for consistency across all replicas.

- Consider the size of your cell and how geographically dispersed the cell is when deciding how many CDS Servers you need. 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.

- Each CDS Server should maintain at least one clearinghouse. All clearinghouses should contain a copy of the root directory, in addition to the other directories that are replicated there.

- Make replication decisions that are based on where the contents of the directories are referenced. Put replicas where the contents are read and put masters where the contents are written.

The following processes comprise the GDA:

- The **gdad** daemon is the GDA server, which sends lookup requests for cell names to either GDS or DNS and returns the results to the CDS clerk in the cell that initiated the request.

- The **gda_child** process is used by **gdad** to communicate with GDS.

In a DCE configuration that uses GDS or DNS, CDS must be able to contact at least one GDA to access global directory service. The GDA can be on the same machine as a CDS Server, or it can exist independently on another machine. You should have at least two **gdad** daemons running in a cell to ensure GDA availability.

4.2.5 DTS Server Programs

The DCE Client configuration already contains all of the files that are necessary for a DTS Server machine, with the exception of the optional time provider. The necessary files are as follows:

- The **dtsd** daemon, which is also installed on a DCE Client machine, is configured to run as a server when installed on a DTS Server machine.

As a server process, **dtsd** synchronizes with other DTS Servers, in addition to synchronizing the local clock, as it does on a client machine.

- The **dts**_*device_name*_**provider** specifies the communications between the DTS Server process and the time-provider process. For *device_name*, substitute the device that you are using, which can be a radio, clock, or modem, or another source of UTC time for DTS. A time provider is optional. If you use a time provider, it must connect to a server process.

Consider the following guidelines when planning your DTS implementation:

- Each cell needs to have at least three DTS Servers. At least three DTS Servers are needed in order to detect if one of them is faulty when they are queried for the time. It is preferable to have four or more DTS Servers to provide redundancy. The additional servers increase the accuracy of time synchronization. However, increasing the number of servers that are queried for the time also increases the activity on the network. The administrator must balance the level of accuracy with the amount of network activity.
- A time provider is optional in DTS; however, cells that must be closely synchronized with a time standard need to have at least one time provider.
- Servers need to be located at the sites with the greatest number of different network connections.

There are many network configuration decisions that affect DTS planning. In the *OSF DCE Administration Guide—Core Components*, you can find details about the total DTS planning process, including configuration planning for Local Area Networks (LANs), extended LANs, and Wide Area Networks (WANs). The *OSF DCE Administration Guide—Core Components* also explains the criteria that you need to use when selecting a time source for your network to use.

4.2.6 GDS Server Programs

A GDS Server machine requires the following files:

admscheme, asn1_attr, common, countries, dirparam, gdscache, gdscacheadm, gdscacheupd, gdschdb, gdscmxl, gdsconf, gdscrontab, gdscstub, gdsdaemon, gdsdbread, gdsdbwrite, gdsdeact, gdsdirinfo, gdsdistcmd, gdsditadm, gdsdsa, gdsexec, gdsgendb, gdshdlcache, gdshdlupd, gdsinfo, gdsipcchk, gdsipcinit, gdsipcstat, gdslanguage, gdslog, gdsmkiss, gdsmkupd, gdssstub, gdsstart, gdsstep, gdssysadm, gdstransfer, gdsutil, ipcconf, newscheme, nsapmacros, osiforminfo

A machine that is configured as a GDS Server runs the GDS Client/Server configuration, which consists of the following three parts:

- Server
  - The **gdsdsa** program is the main DSA program; it forks as many DSA processes as are needed and it accesses the database.
  - The **gdsstub** program is the process that receives incoming requests from clients and responds back to clients; it sends outgoing requests from servers to other servers and receives responses to these requests.
- Client
  - An application, such as **gdsditadm**, links the DUA library.
  - The **gdscacheadm**, **gdscache**, and **gdscstub** programs, which are described in Section 4.1.6, provide additional GDS client functionality.
- Per-machine utilities
  - These utilities are the **gdsipcchk** and **gdssysadm** programs, which are described in Section 4.1.6.

You can have more than one GDS Server (DSA) running in your cell. If you have more than one DSA, the data in the Directory Information Base (DIB), which is the GDS database, can be partitioned by storing a different part of the DIB on each server. Alternatively, the data can be replicated by storing copies of the DIB on several machines. A combination of partitioning and replication can also be used.

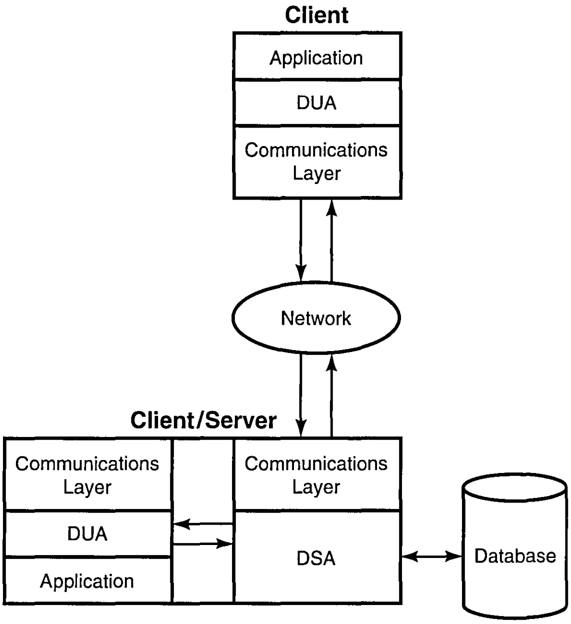
You need to plan what information you want to replicate and how the information is distributed. The master DSA is the only place where writes and updates can occur. Although you can access the master DSA from any client machine, if the master is unavailable no updates can be made. Therefore, you need to choose a dependable machine to run the master DSA. By creating shadows (replicas) of the master DSA you increase the reliability of read operations. By strategically placing shadows on the network you can improve the access time for users.

Keep the following considerations in mind when planning for a DCE configuration that includes GDS:

- The initial installation takes approximately 20 megabytes. One database entry requires approximately 7 kilobytes. You must consider and allocate additional disk space, depending on the amount of information that you want to store in your directory.
- You must understand how CDS cell-related information is entered and displayed using the GDS administration program. (See the *OSF DCE Administration Guide—Extended Services* for a description of the masks (menus) that you use to enter this information.)
- You must know the IP address, port number (together, these two pieces of information are called the PSAP), and the cell name for each machine that has a DSA. (See the *OSF DCE Administration Guide—Extended Services* for information about entering and displaying PSAP addresses using the **gdsditadm** or **gdscacheadm** GDS administration program.)

Figure 4-2 shows a GDS client configuration that goes through the network to access a DSA on another machine that is running the GDS Client/Server configuration.

Figure 4–2. GDS Configurations



4.2.7 DFS Server Programs

DCE supports the configuration of the following types of DFS Server machines:

- DFS private File Server machine
- System Control machine
- File Server machine
- Fileset Location Database machine

The following programs are installed on a basic DFS private File Server machine:

bos, bosserver, dfsbind, dfsexport, epimount, fts, ftserver, fxd, repserver, salvage

The following program is installed on the basic DFS private File Server machine, on AIX only:

epidaemon

For the System Control machine, the following program is added:

upserver

For the File Server machine, the following programs are added:

newaggr, **upclient**

For the Fileset Location Database machine, the following programs are added:

flserver, **newaggr**, **upclient**

The following programs are optional for DFS Servers:

bak, bakserver, butc, cm, fms, repserver, scout, upclient, upserver

DFS File Servers can assume different roles. The DFS space requirements may vary, depending on the role of a particular machine. (See the *OSF DCE Administration Guide—Extended Services* for further information about DFS configuration options.) DFS machines that export data for use in the global namespace can run the following server processes:

- The **flserver** process maintains a complete list of fileset locations in the Fileset Location Database (FLDB). The FLDB is a cell-wide database that maps filesets to the servers on which they are located. There must be at least one **flserver** process running in a cell.
- The **fxd** daemon is a user-space process. The **fxd** daemon starts the kernel processes that implement the File Exporter.
- The **ftserver** process allows system administrators to create, delete, duplicate, move, back up, or restore entire filesets with one set of commands.
- The **bosserver** process reduces system administration demands by constantly monitoring the processes that are running on its File Server machine. The **bosserver** process can restart failed processes automatically; it provides a convenient interface for administrative tasks.
- The **repserver** process manages replicas of filesets on all File Server machines.

- The **upserver** process controls the distribution of common configuration files to all other DFS Server machines in a domain.
- The **upclient** process contacts the **upserver** process to verify that the most recent version of each DFS configuration file is being used.
- The **dfsbind** process is described in Section 4.1.7.

The following text describes the DFS configurations: a System Control machine, a Fileset Location Database machine, a File Server machine, a Binary Distribution machine, and a DFS client that is also a private File Server machine.

A System Control machine distributes system configuration information, such as administrative lists, which is shared by all DFS Server machines in an administrative domain. This machine runs the **upserver** process and the **bosserver** process.

A Fileset Location Database machine runs the **flserver** process. The Fileset Location Database machine tracks the locations of all of the filesets and records the locations of the filesets in the FLDB. The **flserver** process can run on the same machine as the File Server machine.

A File Server machine is used to export DCE LFS and non-LFS data for use in the global namespace. This machine must run the **fxd**, **ftserver**, **bosserver**, and **repserver** processes. File Server machines also run the **upclient** process to receive configuration file updates. The client process, **dfsbind**, must also run on this machine. The full range of fileset operations, including replication, is available on this machine.

Similarly, the Binary Distribution machine stores and distributes DFS binaries for processes and command suites to all other server machines of its Central Processing Unit (CPU) or Operating System (OS) type.

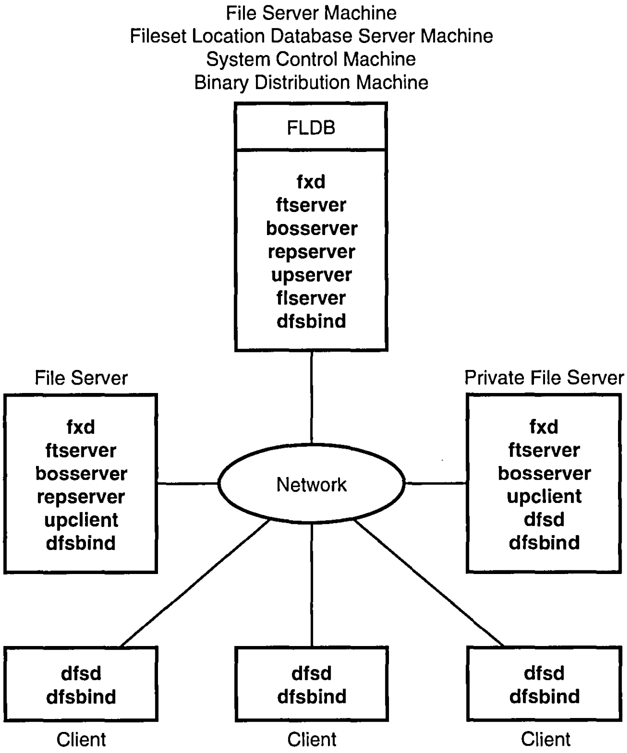
As previously explained, a DFS client machine runs the **dfsd** and **dfsbind** processes. Optionally, a DFS client machine can be configured as a private File Server in order to export its local file system for use in the global namespace. This machine must run the **fxd** and **ftserver** processes. It is recommended that you also run the **bosserver** process.

A private File Server machine is controlled by the owner of the machine, not by the system administrator. The purpose of a private File Server machine is to allow individual users to export a small number of filesets. (See the discussion of DFS client machines in the *OSF DCE Administration Guide—Extended Services* for further information about this configuration.)

Figure 4-3 shows a DFS configuration that uses a File Server machine to run the Fileset Location Database machine, a System Control machine, and a Binary Distribution machine. A second machine is a File Server machine only. One DFS client machine is configured as a private File Server in order to export filesets for use in the global namespace. Note that the first machine is configured to perform multiple roles.

Note: Figure 4-3 shows DFS alone. In addition, each client would run the processes that were previously described in this chapter. A complete cell would also include servers for the minimum DCE configuration.

Figure 4–3. An Example DFS Configuration



The *OSF DCE Administration Guide—Extended Services* provides more information about other DFS configuration options. The *OSF DCE Administration Guide—Extended Services* also describes an additional DFS Server role, the Backup Database machine. A Backup Database machine stores the Backup Database and other administrative information that is used in the DFS Backup System.

4.3 DCE Administration Utilities

The following subsections describe the utilities that assist you in performing DCE administrative tasks.

4.3.1 DCE Administration Space Requirements

You may decide to configure a machine as a DCE Administration machine by installing the DCE administration clients and tools on that machine. The DCE configuration script does not currently support this configuration.

The following programs need to be installed, in addition to the DCE Client software:

RPC Administration Programs:

No software other than the DCE Client software is needed for RPC administration.

CDS Administration Program:

The **cdsbrowser** is needed for CDS administration.

Security Administration Programs:

The **passwd_override**, **sec_admin**, **sec_create_db**, and **sec_salvage_db** programs must be installed for Security administration.

DTS Administration Programs:

No additional software is necessary.

GDS Administration Program:

gdsditadm

DFS Administration Programs:

bak, bos, cm, dfsexport, fts, newaggr, salvage, scout

The following subsections describe the system administration utilities for the DCE components.

4.3.2 RPC Administration Programs

The **rpccp** program is described in Section 4.1.2.

4.3.3 DCE Security Service Administration Programs

The DCE Security Service provides the following administration utilities:

- The **sec_create_db** utility creates the Security database and sets up some configuration files.
- The **sec_salvage_db** command helps you recover from possible program errors and data corruption. The **sec_salvage_db** command applies internal consistency checks to the Security database (registry) and fixes internal data structure problems. You can also use **sec_salvage_db** to generate an ASCII version of the registry that can be edited and reconstructed, if necessary.
- The **acl_edit** command displays, adds, modifies, and deletes ACL entries for a specific object. The *OSF DCE User's Guide and Reference* contains detailed information about using the **acl_edit** command.
- The **rgy_edit** command allows you to edit the Security database or the local registry. Almost all editing of the registry must be done with this command. The *OSF DCE Administration Guide—Core Components* explains the use of the **rgy_edit** command.

- The **passwd_import** command allows you to create registry entries that are based on the group and password files from machines that do not implement DCE Security.
- The **passwd_export** command allows you to update the UNIX **/etc/passwd** and **/etc/group** files with current user information that is obtained from the registry.
- The **passwd_override** file allows you to establish overrides to the information that is contained in the registry.
- The **sec_admin** command helps you to manage server replicas of the registry, change the master server site, and reinitialize a slave server.

4.3.4 CDS Administration Programs

CDS provides the following administration utilities:

- The **cdscp** program, as described in Section 4.1.4.
- The CDS Browser, **cdsbrowser**, is a Motif-based program that lets you view the contents and structure of the CDS namespace.

4.3.5 DTS Administration Programs

The **dtscp** command is the interface that you use to configure and manage DTS. It is already included in the DCE Client software.

4.3.6 GDS Administration Programs

The **gdsditadm** program supports administration of the contents of the local DUA cache database and the GDS database, both local and remote.

4.3.7 DFS Administration Programs

DFS provides the following administration utilities:

- The **salvage** process checks the DCE LFS file system for internal consistency and corrects errors that it finds.
- The **fts** commands help you to manage filesets.
- The **bak** commands help you to perform backup tasks.
- The **cm** commands help you to customize the performance of the Cache Manager and examine the features of DFS.
- The **bos** commands help you to contact the Basic OverSeer (BOS) Server, which is used to monitor processes on server machines in your cell. You can also use the **bos** commands to perform some Security tasks.
- The **scout** program helps you to monitor the File Exporters that are running on File Server machines. You may want to install **scout** only on the system administrator's DFS client machine.
- The **newaggr** command can format a raw disk partition for use as a DCE LFS aggregate.
- The **dfsexport** command makes DCE LFS aggregates and non-LFS partitions available to remote users through use of the File Exporter.

4.4 Application Development Environment Machine

A DCE machine can also be configured for the development of DCE applications. This involves adding to the basic DCE Client configuration several include (**.h**) and interface specification (**.idl**) files, along with the **idl** program.

Chapter 5

Location of Installed DCE Files

This chapter describes the location of DCE files that are created during the installation and configuration processes. The files that are used by DCE are grouped in the following locations:

- The *dceshared* subdirectories
- The *dcelocal* subdirectories
- Conventional UNIX subdirectories

Some information needs to be kept locally on a machine for reliability and to ensure that security is maintained. For example, when you configure DCE, the file that contains the name of your cell must be on the machine that is being configured. This file is stored in the *dcelocal* subtree. Other information that is used in DCE can be and needs to be shared among machines in a cell. For convenience, this information is stored in the *dceshared* subtree.

The *dceshared* subtree is created when you use the **tar** command to retrieve the archived files from the DCE tape, as described in the *OSF DCE Release Notes*. The *dcelocal* subtree is created when you install DCE components, as described in Part 2 of this guide.

The complete set of delivered DCE files, except those that are created during runtime, are stored under *dceshared*. The *dcelocal* subtree is a subset of *dceshared*. Files that are required in conventional UNIX subdirectories and executables that are required in *dcelocal* subdirectories can be duplicates of files and templates that are in the original *dceshared* subtree. In some cases, files are installed into directories such as **/usr/lib**, **/usr/bin**, or **/bin** for performance reasons. In other cases, symbolic links can be used from the conventional UNIX subdirectories to *dcelocal*.

The following sections describe the DCE subdirectories. Appendix B provides the directory layout for *dceshared*, *dcelocal*, and the conventional UNIX subdirectories that are used by DCE.

5.1 The dceshared Subtree

The files in the *dceshared* subtree can be kept on local machines or, preferably, they can be exported to other machines in the DCE cell by using DFS. Therefore, shareable files, including binaries that are addressed by **@sys**, are stored under *dceshared*. The *dceshared* subtree is read-only.

All files that are generated by a DCE build, all files that are delivered to binary licensees, and if appropriate, all files that are delivered to source licensees are initially stored in the *dceshared* subdirectories. All files in the *dceshared* subdirectories are kept unmodified over the lifetime of an installed DCE release. Configuration and data files are only stored as templates in *dceshared*. The actual working set of data files is located in the *dcelocal*/**var** and *dcelocal*/**etc** subdirectories.

The default pathname prefix for *dceshared* is **/opt/dce**, which is a symbolic link to **/opt/dce1.0**, or for DFS, to **/:/opt/dce1.0**, which is the short form of **/.:/fs/opt/dce1.0**. This entry is always physically located at the local machine. Therefore, the local system administrator (or the respective software administrator) must have write permission to modify this link. You can redirect this link from the fileset on the local machine to the cell-wide accessible fileset as soon as the local machine is configured and the cell is available; for example, redirect **/opt/dce** to **/:/opt/dce1.0**.

Note: Special care must be taken because this link is crucial for protecting the local machine if it is running as the client and for protecting the server machine if it is acting as the service provider. This symbolic link must be created in a protected directory, which is comparable to **/etc**. Only the local system administrator needs to have write and modify permissions to this directory.

To avoid having replicas of *dceshared* files on local machines, you can use a symbolic link to access the cell-wide versions of these files. In case DFS users do not want to have replicas of these files physically stored on their local machine, they can remove the *dceshared* subtree that is installed on the local machine and redirect the default symbolic link to the cell-wide *dceshared* subtree, if these particular files are available there. The subdirectory that *dceshared* points to has a version number that is associated with the pathname that provides the capability of running multiple versions of DCE in one cell. This capability is sometimes required in an intermediate phase of upgrading to a new release. An additional advantage is a simplified deinstallation procedure.

If necessary, you can create copies or symbolic links from the other locations to **/opt/dce/***, such as *dcelocal*/**var**, *dcelocal*/**etc**, *dcelocal*/**bin**, and **/usr/bin**. These guidelines are based on the assumption that you want to use the DCE capability of cell-wide file sharing. The pathname for *dceshared* is set at compile time and is not associated with any particular version number.

5.2 The dcelocal Subtree

In order to initially boot a server and configure the cell, the appropriate files for mandatory servers (CDS and Security) need to be available on that server machine (in the *dcelocal* subtree). It is strongly recommended that copies of the minimum set of programs and data files that were installed during the default DCE installation procedure be kept locally on server machines for standalone operation and emergency maintenance.

The contents of the *dcelocal* subtree can vary from machine to machine inside a DCE cell to accommodate and serve specific configurations. In addition, every machine must have local access to certain files so that each machine can run as a standalone system if the machine is disconnected or partitioned from the cell. The appropriate files on DCE servers that have to be local to the server machine must be stored under *dcelocal*. Client-related data files are stored under *dcelocal*/**etc** (static configuration data) and *dcelocal*/**var/adm** (log files and so forth). All server-specific data files are located in the *dcelocal*/**var**/*dce-component-name* directory.

The default pathname for *dcelocal* is set to **/opt/dcelocal** during the configuration process. This is a fixed pathname. Every machine must have local access to the files that are necessary to configure it, up to activating DFS access in the cell. The **/opt/dcelocal/dce_cf.db** file is the DCE configuration file that contains the name of the host that is to be configured and the cell name. A machine must access this small set of DCE files, which is kept on the machine's local disk, to start up the various DCE components and for local configuration information and log information.

Because DCE configuration takes place after mounting the local file systems, none of these files has to be available in the root partition, except for DCE Diskless Support Service. If diskless operation is supported, a few files (for example, **dfsd**) must have copies in the root partition (**/sbin**).

During DCE configuration, only the executables in *dcelocal*/**bin** are reliably available. Start-up procedures, such as **rc** scripts, need to address executables through *dcelocal*/**bin** rather than **/usr/bin**, even if the same files are believed to be in both directories. Commands in **/usr/bin** can be just symbolic links to *dcelocal*.

The *dcelocal* subtree is populated and initialized during DCE configuration.

5.3 Conventional UNIX Directories

Some files and directories that are used by DCE are accessible in conventional UNIX directories. These DCE files and directories need to be accessible in conventional locations so that users can conveniently access frequently used utilities and data, such as **idl** from the **/usr/bin** directory and **localtime** from the **/etc/zoneinfo** directory. Header files are accessible in **/usr/include** or in its subdirectory, **/usr/include/dce**, and in libraries, such as **libdce.a**, are kept in **/usr/lib**.

5.4 UNIX Permissions for DCE Subdirectories

All directories in the file system are created with the UNIX permissions set to **rwxr-xr-x** for user **root** and group **bin**. In subsequent configurations, the DCE Security Service can define the roles for several administrators (principals or groups). A possible scenario follows:

- A software administrator who owns the installed software packages and has write and modify permissions for the entire set of files included in *dcelocal*.
- DCE service administrators who are responsible for a particular DCE service such as Security and have read and write permissions for the data files for the respective service. You can assign a separate DCE Security Service administrator, while a single cell administrator can have responsibility for the remaining DCE services.
- A local DCE system administrator who is responsible for client setups and has read and write permissions for the respective local files.

Chapter 6

Overview of DCE Maintenance

Once you have performed the tasks that are required for planning, installing, and configuring your DCE system, you can go on to perform the tasks that are required for maintaining the system. The initial tasks of planning, installing, and configuring are performed infrequently, some only once. Maintenance tasks, however, are performed on a regular basis throughout the lifetime of your system.

Maintenance of a distributed system includes the following areas:

- Performance tuning
- Configuration control
- Security and access control

This chapter summarizes some of the primary DCE system administration tasks. The first section of this chapter tells you how to start up DCE. The remaining sections describe tasks that apply to the individual components of DCE. DCE component tasks are documented in detail in the *OSF DCE Administration Guide—Core Components* and the *OSF DCE Administration Guide—Extended Services*.

6.1 Starting Up DCE

The **dce_config** script, which is described in Chapters 7, 8, and 9, creates an **/etc/rc.dce** file that is used to start up the DCE processes. This file is customized to an individual machine's configuration. To start up the DCE processes, enter the following command:

```
sh /etc/rc.dce
```

6.2 CDS Maintenance Tasks

CDS components, including clerks, servers, and clearinghouses, are largely self-regulating. Except for routine monitoring, CDS requires little intervention for system administration. When intervention is required, CDS provides system administration tools to help you monitor and manage the CDS namespace and CDS Servers.

You can use the **cdscp** commands to create and manage the components of a CDS namespace. The **cdscp** program is an interactive or command-driven management interface.

You can also manage CDS by using the CDS Browser utility to view the namespace. The CDS Browser enables you to monitor growth in the size and number of CDS directories in your namespace. You can use the CDS Browser to display an overall directory structure, as well as the contents of directories.

If you have a large organization, you can improve efficiency by having one system administrator responsible for CDS Servers and another system administrator responsible for the namespace. You can delegate responsibility for a subtree of the namespace to another administrator by granting access control rights to that person.

(See the *OSF DCE Administration Guide—Core Components* for more detailed information on CDS maintenance tasks.)

6.2.1 Monitoring CDS

CDS monitoring tasks fall into the following two categories:

- Monitoring the namespace
  - Monitor the size and usage of clearinghouses and determine the need for new CDS Servers and clearinghouses. Plan and oversee the configuration of these new servers and clearinghouses.
  - Maintain and monitor a map of the namespace.
- Monitoring CDS Servers
  - Enable event logging, monitor CDS events, and solve system-specific problems if they arise. If necessary, notify the namespace administrator of problems that can affect other CDS Servers or clerks.
  - Monitor the success of skulks that originate at the server. A *skulk* is a method of updating all replicas through repeated operations.
  - Monitor the size and usage of the server's clearinghouse and, if necessary, discuss with the namespace administrator the need to relocate some replicas or create a new clearinghouse.
  - Monitor and tune system parameters that affect or are affected by CDS Server operation.

Note: When monitoring memory usage for CDS clerks, it is important to understand that memory remains allocated under certain conditions. Memory that is associated with objects remains allocated until a skulk is successfully completed. Memory that is associated with directories remains allocated until the server is disabled and restarted.

6.2.2 Managing CDS

CDS management tasks fall into the following two categories:

- Managing the namespace
  - Oversee the creation of new directories and assign names according to a standard, or enforce established guidelines in the assigning of

names. Note that beyond a certain level in the directory hierarchy, you can delegate the responsibility of creating and maintaining directories. You need to keep track of the new directories that are being created to make sure they are appropriately replicated.

  - Determine the default access control policy.

  - Administer and enforce the established access control policy for directories and entries.

  - Determine where and when new replicas of a directory are necessary.

  - Create soft links for objects that change locations or for objects that need to be renamed. An *object* is a resource, such as a disk, an application, or a node, that is given a CDS name. A name plus its attributes make up an *object entry*. A *soft link* is a pointer that provides an alternate name for an object entry.

    Publicize and encourage the use of the new names so that eventually the soft links can be deleted.

  - Solve or direct the resolution of problems involving multiple CDS Servers.

- Managing CDS Servers

  - Manage access control on directories and objects, and monitor the size and usage of directories in the server's clearinghouse. Create new directories, possibly with the namespace administrator, when necessary.

  - Create new objects in directories or oversee their creation. Note that beyond a certain level in the directory hierarchy, you also can delegate the responsibility of maintaining directories and the objects in them.

  - Add new administrators to the **cds-admin** security group.

6.2.3 CDS Security and Access Control

The CDS ACL Manager and the DCE ACL Editor (**acl_edit**) work together to manage authorization in CDS. The CDS ACL Manager is an integral part of the **cdsd** and **cdsadv** processes. When a **cdscp** request is issued to perform an operation on a CDS object, the CDS ACL Manager checks permissions, based on ACL entries, and grants or denies the request. To modify, add, delete, or view ACL entries in the CDS namespace, use the **acl_edit** command.

The *OSF DCE Administration Guide—Core Components* provides detailed information about handling CDS security and access control, including guidelines for setting up access control in a new namespace.

6.3 GDS Maintenance Tasks

GDS provides a menu-driven interface for performing the maintenance tasks described in the following subsections. (See the *OSF DCE Administration Guide—Extended Services* for more detailed information on the GDS maintenance tasks.)

6.3.1 Monitoring GDS

You can monitor GDS by displaying directory system status information with the GDS interface or by using the trace system to log directory processes.

The GDS interface allows you to do the following:

- Display directory system status information, which shows if a directory is active or inactive, what processes are available, how many processes are available, and if the trace system is active or inactive.
- Activate the trace system, which starts the trace system for logging directory processes.
- Deactivate (or stop) the trace system.

GDS maintains log files for each of the following processes:

- The DUA process
- The Cache process
- The Client-stub process
- The Server-stub processes
- The DSA processes
- The GDS system administration process
- The Monitoring process

6.3.2 Managing GDS

GDS provides the following functions for management tasks:

- `Administrative Functions`

  This directory management function, which is available after logging into a DSA, supports administration of the contents of the GDS database, both local and remote. You can choose from the four types of administration functions in the following list. These administration functions allow you to administer objects, shadows, trees, and schema for your directory service system. A GDS *schema* is a set of rules and constraints for tree structure, object class definitions, attribute types, and syntaxes that characterize the Directory Information Tree (DIT).

  — `Object Administration`

  This function controls objects and changes their attributes.

  — `Schema Administration`

  This function modifies the Object Class Table, Structure Rule Table, and Attribute Table to store a new schema in the Directory System Agent.

  — `Shadow Administration`

  This function executes, schedules, or modifies shadow jobs for updating.

  - `Tree Administration`

    This function adds, deletes, or modifies subtrees.

- `Administrative Functions Under the DUA Cache`

  This directory management function supports the administration of the contents of the local DUA cache database. You can choose from the two types of administration functions in the following list. These administration functions allow you to administer the objects that are stored in the local DUA cache database and the cache update job.

  - `Object Administration`

    This function controls objects that are stored in the local DUA cache database and changes their attributes.

  - `Cache Update`

    This function displays, activates, or deactivates the Cache Update job, or changes its update frequency.

- `Activation of a directory system installation`

  This directory management function activates the directory installation by starting the background processes of GDS.

- `Deactivation of a directory system installation`

  This directory management function ends the background processes of GDS. All running directory processes are ended, but running operations are not interrupted, and the data consistency of the managed data is retained.

(See the *OSF DCE Administration Guide—Extended Services* for more information on these directory management functions.)

6.3.3 Backing Up GDS Data Files

You can back up directory system data files in GDS by selecting the entry that saves the local data files to diskette or tape from the function entries in the menu mask. The save process backs up all the local data files (local DSA data, DUA cache data) that belong to your directory system.

GDS has a password feature that protects the directory system data files on the data medium. The use of this password is optional when saving data, but you must use the password when loading files that are saved with a password.

6.3.4 Changing Global Directory Configurations

The GDS interface has a function, `Configuration of a directory system`, that enables you to enter, delete, display, or change configuration data, such as the number of clients or servers that are to be activated.

6.4 DTS Maintenance Tasks

Like CDS, DTS is largely self-regulating once configuration of the service is complete. However, there are times when you need to intervene. Use the **dtscp** command to perform the following DTS configuration and management tasks:

- Identify system clock problems.
- Adjust the system clocks.
- Change DTS attributes for varying WAN conditions.
- Modify the system configuration when the network environment changes.

(See the *OSF DCE Administration Guide—Core Components* for more detailed information on DTS maintenance tasks.)

6.4.1 Managing the Distributed Time Service

You can use the **dtscp** command interface to create and enable DTS. Once this is done, you can perform routine management tasks, such as enhancing performance, reconfiguring the network, and changing local time.

Several commands and characteristics modify and improve the performance of your network. The **set** command changes the values of many of these characteristics. The **show** command displays the values of characteristics at any time. The following are some of the tasks that you can accomplish by using the DTS commands and the characteristics of DTS that can be set:

- Display or change the number of servers that must supply time values to the system before DTS can synchronize the system clock.
- Display or change the inaccuracy limit that forces the system to synchronize in order to bring the inaccuracy back to an acceptable level.
- Display or change the interval at which you want clock synchronization to occur.
- Display or change the reaction to a faulty system clock.
- Display or change the settings that indicate how often to query servers.

Refer to the *OSF DCE Administration Guide—Core Components* for more information on these and the following tasks:

- Creating and enabling DTS.
- Assigning the courier role to servers to facilitate communications to other parts of your network.
- Matching the epoch number for servers that you add to your network after the initial configuration. An *epoch number* is an identifier that a server appends to the time values it sends to other servers. Servers only use time values from other servers with whom they share epoch numbers.
- Advertising DTS Servers to CDS, thereby registering them as objects in the namespace.

6.4.2 Modifying System Time

Sometimes you need to modify the system time. You can update time to match the international time standard, Coordinated Universal Time (UTC), from a source such as telephone, radio, satellite, or another external referencer, if your network does not use time providers and the network systems have been running for some time. The **update** command accomplishes this task by gradually modifying the time.

The **change** and **synchronize** commands provide additional methods for adjusting the system clock and synchronizing systems.

6.5 DCE Security Service Maintenance Tasks

The following subsections summarize the maintenance tasks that you perform while administering the DCE Security Service. (See the *OSF DCE Administration Guide—Core Components* for more detailed information on Security maintenance tasks.)

6.5.1 Managing the DCE Security Service

The DCE Security Service management tasks include the following:

- Creating and maintaining accounts by using the registry editor

  The **rgy_edit** command is a structured editing interface that is used to create and maintain registry information, including persons, groups of users, and accounts.

  Keep the following considerations in mind when administering DCE accounts:

  — If you share files with other systems that do not use the registry, be sure that names, UNIX IDs, and account information are consistent between the registry and the foreign password and group files. Use **passwd_import** to identify and resolve any conflicts that exist.

  — If you maintain **/etc/passwd** and **/etc/group** files in standard UNIX format, you need to run **passwd_export** to make password, group, and organization files on local machines consistent with the registry.

  — For principals in other cells to access objects in your cell, you need to set up a special account for the foreign cell in your cell's registry. This account indicates that you trust the Authentication Service in the foreign cell to correctly authenticate its users. Use the **rgy_edit cell** subcommand to create an account for a foreign cell.

- Using ACLs

  Use the **acl_edit** command to display, add, modify, and delete ACL entries for a specific object in the cell namespace. (See the *OSF DCE User's Guide and Reference* for detailed information on how to use the **acl_edit** command.)

- Setting and maintaining registry policies

  Registry policies include certain password and account information. Policies also include overrides, which are exceptions that are tied to a specific machine. Use the **rgy_edit** command to set and maintain registry policies.

  Ticket expiration date, password life span, password format, and password expiration date are examples of registry policies that you can set. If both an organizational policy and a registry policy exist for password format, for example, the more restrictive policy applies.

  You can establish overrides to the information that is contained in the registry. Override information is stored in the **passwd_override** file on a local machine. The **passwd_override** file contains the home directory, the login shell, entries for overriding the password, and GECOS information, which is general information that is used by users but not required by the system, such as office and phone numbers.

- Backing up the registry

  The *OSF DCE Administration Guide—Core Components* describes the back-up procedure to follow for the master registry site. When you restore the database, it is automatically propagated to the slaves.

- Troubleshooting

  When you encounter problems that cannot be resolved through routine management procedures, or when hardware failures stop the registry from operating, there are several troubleshooting procedures you can use. The *OSF DCE Administration Guide—Core Components* describes the following tasks:

  - Recreating a registry replica
  - Recovering the master registry
  - Forcibly deleting a replica
  - Adopting registry objects that are orphaned because their owner has been deleted

6.5.2 Reconfiguring the Registry

There are two main reconfiguration tasks included in the administration of the DCE Security Service. The following tasks are described in the *OSF DCE Administration Guide—Core Components*:

- Changing the master registry site when you plan to move the machine that runs the master registry server from your network or shut the machine down for an extended period
- Removing a server host from the network when you plan to remove a machine that runs a slave registry server from the network or shut that machine down for an extended period

6.6 DFS Maintenance Tasks

The following subsections summarize the five major DFS maintenance tasks: monitoring DFS servers and clients, managing filesets in a cell, backing up filesets, reconfiguring the Cache Manager, and managing DFS security. (See the *OSF DCE Administration Guide—Extended Services* for more detailed information on DFS maintenance tasks.)

6.6.1 Monitoring DFS Servers and Clients

You can monitor DFS in the following ways:

- Use the BOS Server to continually monitor DFS server processes. You define which processes the BOS Server monitors, and you control server process status by issuing **bos** commands to perform routine maintenance or to correct errors in addition to those that the BOS Server handles. You can also use **bos** commands to restart DFS weekly.
- Use the **scout** program to monitor the File Exporter, which runs on File Server machines. The File Exporter makes DFS files available to client machines. The **scout** program collects and displays information about the machines that you select to monitor. It displays information such as disk usage and the number of connections a machine has.

6.6.2 Managing Filesets in a Cell

The basic unit of administration in DFS is the fileset, which is a collection of related files. The *OSF DCE Administration Guide—Extended Services* describes the following tasks:

- Creating read/write filesets
- Replicating filesets
- Creating backup filesets
- Mounting and naming filesets
- Listing information about filesets
- Moving filesets
- Examining the FLDB entry
- Salvaging filesets
- Synchronizing fileset information
- Setting and listing fileset quota and current size
- Removing filesets and their mount points
- Dumping and restoring filesets
- Renaming filesets
- Unlocking and locking FLDB entries

6.6.3 Backing Up Filesets

The system administrator uses the Backup System that is provided by DFS to make backup tape copies of filesets. For a discussion of how often to perform backups using the **bak** commands, which filesets need to be backed up, and when to make full or incremental backups, refer to the *OSF DCE Administration Guide—Extended Services*.

The *OSF DCE Administration Guide—Extended Services* also describes the following tasks:

- Configuring a backup machine
- Installing tape coordinators
- Listing information from the backup database
- Creating and reading tape labels
- Performing a backup
- Performing a restore
- Canceling backup and restore operations

6.6.4 Reconfiguring the Cache Manager

Usually, all Cache Manager machines are configured in the same way, but you can change certain features to achieve different levels of performance across client machines. You can use the **cm** commands to perform the following tasks:

- Directing the Cache Manager to use machine memory instead of disk space for caching
- Changing the cache size
- Changing the cache location
- Altering the default size and the numbers of chunks that compose a cache
- Directing the Cache Manager to allow programs that reside in foreign cells to execute with **setuid** status
- Changing the cell membership
- Forcing the Cache Manager to discard or fetch a new version of a file or directory from the File Server machine

6.6.5 DFS Security and Access Control

In DFS, you can set up administrator groups with special privileges that permit members of a group to do the following:

- Issue administrator commands
- Create or remove filesets
- Perform system backups

In DFS, administrative lists define the principals that can perform actions affecting specific server machines. Use the **bos** commands to create and maintain administrative lists. Use the **rgy_edit** command to create administrative groups and to place these groups on administrative lists. Adding and removing users from groups rather than altering the administrative lists themselves simplifies system administration.

Groups of DFS server machines that are administered as a single unit are known as DFS administrative domains. Whenever you add or remove server machines in a DFS domain, you must also alter the keytab file for that machine. A keytab file contains a server encryption key, which is used to provide security between servers and their clients. Use the **bos** commands to maintain a server's keytab file.

To verify or modify ACLs, use the **acl_edit** command.

6.7 DCE Diskless Support Service Maintenance Tasks

Maintenance of the DCE Diskless Service involves the following tasks:

- Managing the diskless boot process

  You must set up the boot protocol program (**bootpd**) and the Trivial File Transfer Protocol (TFTP) program (**tftpd**) on the appropriate boot server hosts in the DCE cell, creating the **bootptab** configuration file for each server, and adding entries to the standard **inetc.conf**, **password**, and **services** files.

- Managing diskless configuration

  You manage the configuration of operating systems on diskless clients through an entry in the Diskless Configuration (DLC) Server database. Use the **dlctab** configuration file to maintain the definitions and values of the diskless configuration data. The **dlcd** daemon reads that file and creates the diskless entry and the data subentries that are returned to a diskless client during initialization.

- Configuring and managing the client's root file system

  The configuration of operating systems on diskless client hosts is managed through an entry in the Diskless Configuration Server database. The diskless client uses information that is stored in the DLC database to establish communications with DFS, to set up its file cache in memory, to mount its root file system, and to communicate with the remote swap server, if necessary.

- Configuring the diskless swap server

  For a host with local storage to function as a swap server for one or more diskless clients, the host must run the **dswd** swap server daemon, which is configured with the **dsw_adm** commands.

(See the *OSF DCE Administration Guide—Extended Services* for more detailed information about DCE Diskless Support Service maintenance tasks.)

Part 2

Configuring and Starting Up DCE

Chapter 7

Overview of the DCE Installation and Configuration Script

Part 2 describes the installation and configuration of DCE as implemented by the **dce_config** program, which is provided by OSF with the DCE offering. Your system vendor may provide alternative installation and configuration tools; if so, refer to your vendor's documentation for that software. The DCE installation and configuration script is based on technical considerations and is not tailored to DCE packaging.

You can use the DCE installation and configuration script (''the script'' for short) to bring up DCE Clients and DCE servers. The script is installed in **/etc**. It provides a menu-driven interface for bringing up DCE on a machine.

The script can be used to perform the following actions:

- Creating the required DCE subdirectories on the local machine
- Installing only those files that are required to customize the configuration of this machine
- Setting up the first Security and CDS Servers in your cell
- Adding DTS, GDA, and DFS Servers, and additional CDS Servers to your cell

- Configuring DCE Client machines
- Installing the DCE Application Development Environment

The installation and configuration script can install GDS programs. (See the *OSF DCE Administration Guide—Extended Services* for specific information on configuring GDS.)

You can only use the script for initial DCE installation and configuration tasks. The script does not reconfigure DCE once it is up. (See the *OSF DCE Administration Guide—Core Components* and the *OSF DCE Administration Guide—Extended Services* for information on customizing and reconfiguring DCE servers.)

The script provides a series of menus that guide you through the details of configuring each type of DCE service. Briefly, the configuration procedure consists of three steps:

1. The script creates the required directories and configuration files, Security principals, and namespace entries.
2. Then, the script starts the required daemons.
3. Finally, the script creates an **/etc/rc.dce** script to start up DCE daemons at system initialization.

You must install and configure the first few DCE servers in a specific order. A Security Server must be configured first. Then, a CDS Server must be configured. You can install and configure other servers in any order, but all servers should be configured before any DCE Clients are configured.

You can modify the script to meet your needs, depending on the number of machines that are available, their disk space limitations, and which DCE services you intend to use.

This chapter tells you how to use the script's menus to bring up and configure DCE, and it describes what the script does. Read the entire description in this chapter before beginning the initial configuration of DCE on your system.

7.1 Prerequisites

Before you begin the initial DCE configuration process, be sure that you have

- Completed the installation instructions that are contained in the *OSF DCE Release Notes*
- Conformed to the disk space requirements that are outlined in Chapter 4 of this guide and in the *OSF DCE Release Notes*

In addition, note which of the following subtopics apply to your system, and make any necessary adjustments.

7.1.1 Ethernet Addresses

Some DCE services (namely, RPC and CDS) require a unique number for each machine that is running in the cell. DCE uses a 12-digit hexadecimal number that is reigstered with IEEE as an Ethernet address. If your machine has an Ethernet card, use that address. Some operating systems (for example, the RIOS) provide routines to access the Ethernet address from the card. Others (for example, OSF/1) do not. On an OSF/1 machine, you must therefore store the 12-digit hexadecimal Ethernet address in the following file:

/etc/ieee_802_addr

The format of the file is the 12 hexadecimal digits that are written in text; for example:

```
08002BFFFFFF
```

If your machines do not already have Ethernet addresses, you must obtain them before bringing up DCE.

7.1.2 Locations of Message Catalogs

The script installs the message catalogs during the installation pass. It computes a default location for these catalogs, which is based on the values of the environment variables **NLSPATH** and **LANG**. It prompts you to verify the default location or to change it. During the configuration and start-up portions of the script, these variables must be set correctly for the DCE daemons to work properly.

Your system default for **NLSPATH** will be used by the configuration script; you should not need to change this.

Set your **LANG** environment variable to the correct language (it may already be set for you).

On an IBM RS/6000 that is running AIX:

```
export LANG=En_US
```

On OSF/1 systems:

```
export LANG=en_US.ISO8859-1
```

You can use the **locale** command to verify your environment.

7.1.3 Wide Area Network Connections

In order to have WAN communications within a cell or between cells, you must open the following ports for receiving packets on both ends of the WAN connection:

- The **udp** port 88 for Kerberos
- The **udp** port 135 for **rpcd**
- The **tcp** port 135 for **rpcd**
- All **udp** and **tcp** ports greater than 1024 for all DCE services and applications

In order to open these ports, you may need to program routers at each end of the WAN. (See Section 9.1 for information on configuring a DCE Client over a WAN.)

7.2 Examples of Names Defined by the Script

The script uses the names that are listed in Table 7-1. These values coincide with the default names that are used by the code. The script uses the cell namespace that is described in Appendix A. It is recommended that you do not change these namespace names.

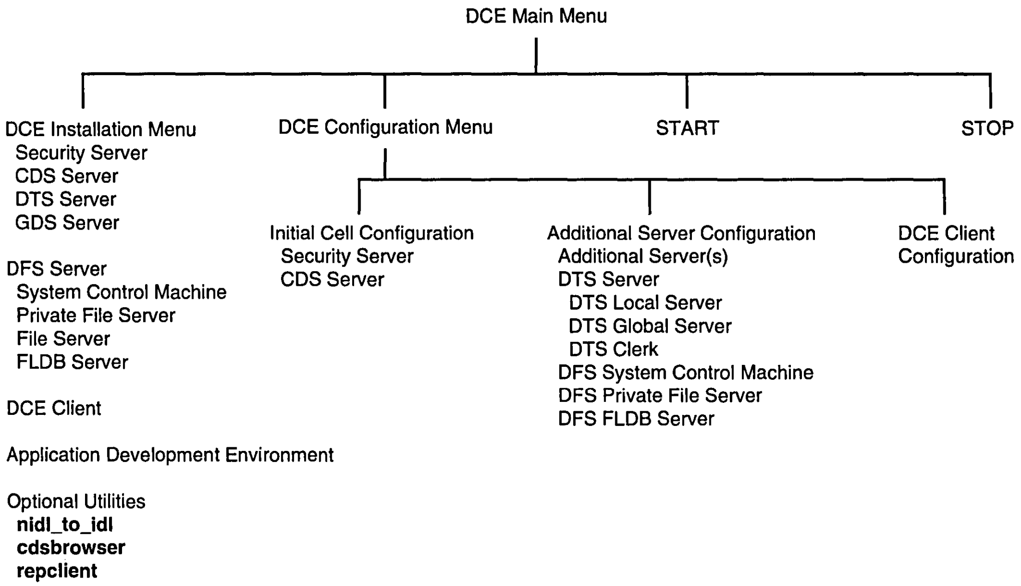
Table 7–1. Names Defined by the Installation and Configuration Script

Name	Value
DCEROOT	/opt
DCELOCAL	/opt/dcelocal
DCESHARED	/opt/dce
SUBSYSDIR	/subsys/dce
SECURITYDIR	/subsys/dce/sec
DFSDIR	/subsys/dce/dfs

7.3 Choices You Need to Make

The menus that guide you through DCE configuration consist of a main menu and submenus for the installation and configuration. A map of the menus and the possible selections on those menus is shown in Figure 7-1. The menus and their selections are described in more detail in the following subsections.

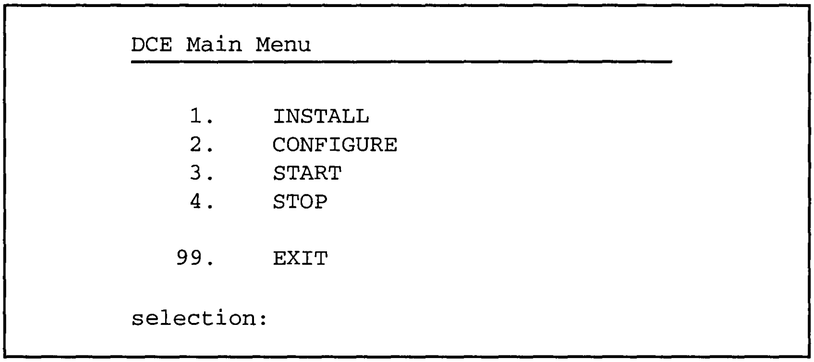
Figure 7–1. DCE Configuration and Installation Menus



7.3.1 Main Installation Menu

The `DCE Main Menu` (see Figure 7-2), which appears when you first invoke the script, lets you select the basic actions that are involved in bringing up DCE on your machine(s). The script reverts to the main menu when transitioning from installation to configuration, and when you start or stop DCE.

Figure 7–2. DCE Main Menu



```
DCE Main Menu

    1.     INSTALL
    2.     CONFIGURE
    3.     START
    4.     STOP

   99.     EXIT

selection:
```

When you choose INSTALL from the DCE Main Menu, a new screen comes up and asks you whether the DCE binaries that you are installing are located on a file system or a media device.

Note that the install tree is not shipped on the DCE tape; the *OSF DCE Release Notes* tell you how to build and install the DCE binaries.

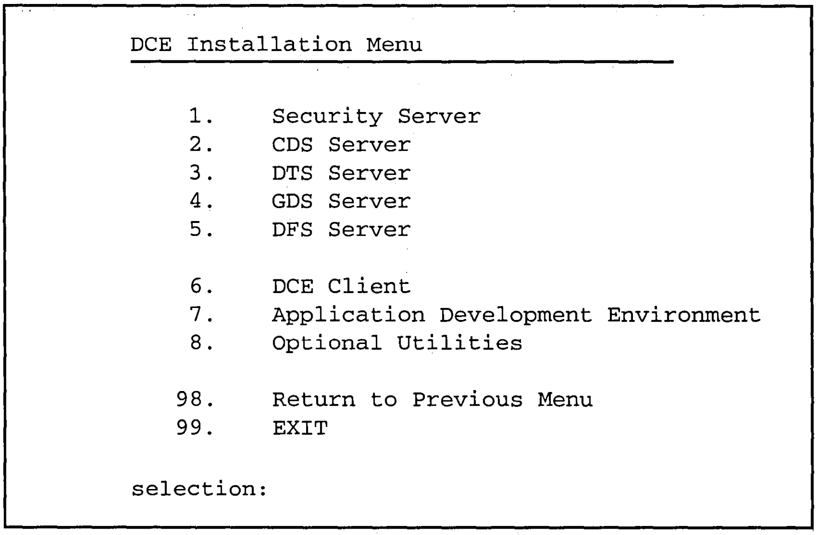
- For a file system installation, enter the pathname to the DCE install tree. If you install from a file system, you must have access to a copy of the DCE install tree, either on the local machine or by a remote mount.
- To install from media, you must create a DCE install tree on the distribution media. To create the media, use the following commands on a machine that contains a complete copy of the DCE install tree:

  cd .../install/*machine_type***/opt**
  tar -cvf *device* **dce1.0.1**

  You can use the resulting media with the installation and configuration script. Since the script uses the **tar** command to retrieve archived files when restoring from the media, it is not normally as fast as installing from a file system.

The DCE Installation Menu is shown in Figure 7-3.

Figure 7–3. DCE Installation Menu



```
DCE Installation Menu

     1.    Security Server
     2.    CDS Server
     3.    DTS Server
     4.    GDS Server
     5.    DFS Server

     6.    DCE Client
     7.    Application Development Environment
     8.    Optional Utilities

    98.    Return to Previous Menu
    99.    EXIT

selection:
```

When you select an option from the DCE Installation Menu, only those binaries that are required for that option are stored in **/opt/dcelocal.** This selectivity lets a minimal set of binaries be stored on each machine, while providing all of the necessary functionality.

Since all machines are DCE Clients, the binaries required for a client configuration are installed on every machine.

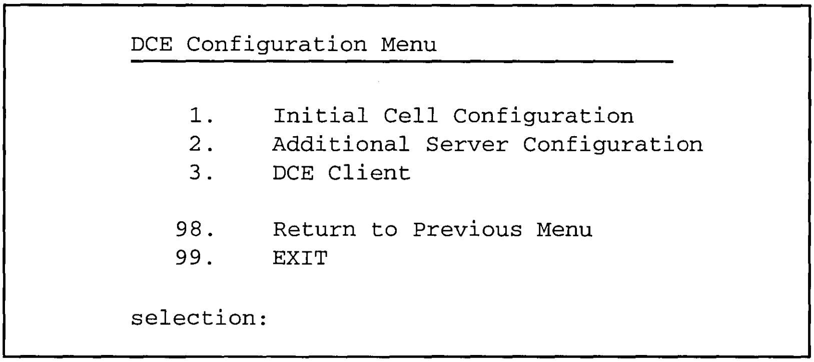
If you select the Application Development Environment option from the DCE Installation Menu, the header files and the IDL compiler are installed. If you select the Optional Utilities option, you get a new menu, which lets you install the following utilities:

- **nidl_to_idl**
- **cdsbrowser**
- **repclient**

7.3.2 Configuring DCE

Once you have installed DCE on your machine, you can proceed to the `DCE Configuration Menu` (see Figure 7-4).

Figure 7–4. DCE Configuration Menu



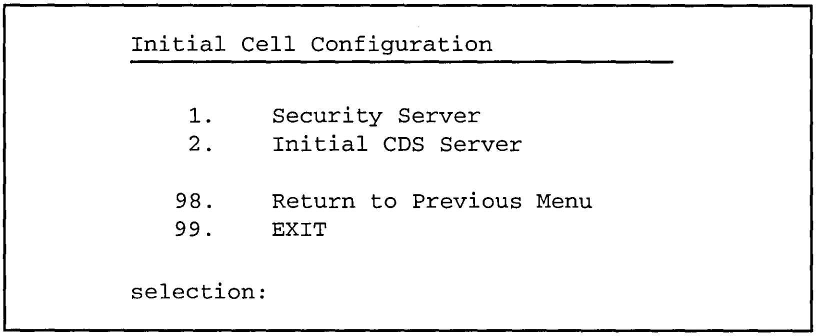
```
DCE Configuration Menu

    1.    Initial Cell Configuration
    2.    Additional Server Configuration
    3.    DCE Client

   98.    Return to Previous Menu
   99.    EXIT

selection:
```

You must start by configuring a Security Server; this step is available under a submenu of the `Initial Cell Configuration` option.

7.3.3 Initial Cell Configuration

To configure the Security Server and the first CDS Server, select the `Initial Cell Configuration` option (see Figure 7-5) from the `DCE Configuration Menu`.

Figure 7–5. Initial Cell Configuration Menu



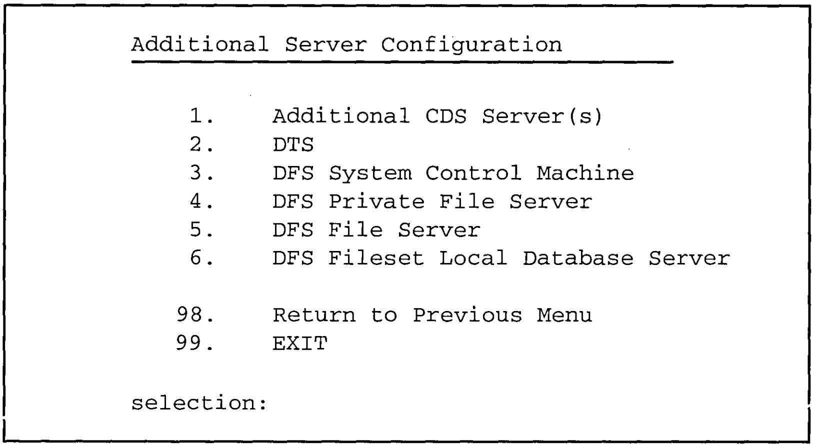
```
Initial Cell Configuration

    1.     Security Server
    2.     Initial CDS Server

   98.     Return to Previous Menu
   99.     EXIT

selection:
```

7.3.4 Configuring Additional Servers

After configuring a Security Server and the first CDS Server, go to the `Additional Server Configuration` menu (see Figure 7-6). From this menu, you can configure additional CDS Servers, DTS Servers, various DFS Servers, and additional Security Servers.

Before you finish configuring servers, you must install a DTS Server on some machine in the cell.

Figure 7–6. Additional Server Configuration Menu



```
Additional Server Configuration

     1.    Additional CDS Server(s)
     2.    DTS
     3.    DFS System Control Machine
     4.    DFS Private File Server
     5.    DFS File Server
     6.    DFS Fileset Local Database Server

    98.    Return to Previous Menu
    99.    EXIT

selection:
```

The DTS option lets you configure a local DTS Server, a global DTS Server, or a DTS clerk. If your cell contains more than one LAN, you need a global server.

If your cell uses more than one LAN, you must configure a separate LAN profile for each LAN with which the cell connects. The script does not handle such configurations, so you must configure these profiles manually. Instructions for doing so can be found in Section 8.5.

When you configure a DCE server on a machine, the machine is automatically configured as a DCE Client. After you configure all of the servers in the cell, select DCE `Client` to configure a DCE Client on each machine that is not a server. You also use the DCE `Client` option to configure a DTS clerk, or to change a DTS Server back to a DTS clerk.

7.4 Customizing the Script

The default value for *cell_administrative_user* is **cell_admin**. If you want to change this value, you must modify the script before configuring the Security Server. You can change the following default values, among others:

- The cell administrator's name and password: use the *celladmin* and *cellpw* shell variables.

  If you change the password for *celladmin* in the Security registry database to a new password, you must also change the value of *cellpw* in the script to the new password in order for the script to run correctly.

- The values for **DFS/LFS**: use the **LFS_PARTITION**, **LFS_AGGID**, and **LFS_AGGNAME** shell variables.

- The values for the **DFS/JFS** shell variables, such as **JFS_ROOTNAME** and **JFS_ID**.

- Values such as the following can also be changed:

  — **RAM_CACHE_SIZE**
  The size of the RAM cache, in kilobytes, of the DFS RAM cache for configuring the DFS Cache Manager to use an in-memory cache.

  — **DISK_CACHE_DIR**
  The location of the local DFS Client cache. Normally, this location is **/opt/dcelocal/var/adm/dfs/cache**.

  — **DISK_CACHE_SIZE**
  The size of the local DFS Client cache. Normally, this value is 10,000 kilobytes.

  — **DFS_THREADS**
  How many simultaneous threads **fxd** (the DFS File Exporter kernel daemon) can support. The default number is 7.

  This value can be tuned for optimal performance. In general, a small number slows down I/O, but a large number can cause **fxd** to run out of resources. Since **fxd** is a kernel daemon, you can only update the value of **DFS_THREADS** by editing the script and rebooting the machine.

— **DFS_SERVERS**
Where the DFS object resides in the file system; normally, it resides in **hosts/$HOSTNAME/dfs-server**.

— **DFS_ADMIN_GROUP**
The DFS Security group. Normally, the group is *dce_root_dir*/**subsys/dce/dfs-admin**. A DFS administrator must be in the Security group to be authenticated.

The location of the **bosserver** configuration file, namely **BOSCONFIG**, cannot be changed. It is located in **/opt/dcelocal/var/dfs/BosConfig**.

7.5 Running the Script

You must be **root** to bring up DCE. If you are installing from a media device, the script must first be restored from the media with the following command:

tar -xvf *media device* **dce1.0.1/etc/dce_config**

To begin an installation, enter the following command from the directory containing the script:

dce_config

The script offers menus for selecting actions. You can bring up any DCE servers with the same script.

On machines that are running OSF/1 Release 1.1.1, you must invoke the script as an argument to **ksh** for the return function to work correctly. Enter:

ksh dce_config

Note: The script itself does very little error checking. If you do not watch the screen closely, error messages may appear on it, and then be cleared by the appearance of a new menu. You may think that a particular step succeeded when it did not.

Proceed to Chapter 9 of this guide for specific information on configuring DCE Clients and the various servers.

7.6 Error Recovery

If a problem occurs sometime during the installation and configuration process, perform the following three steps to return your system to a consistent state:

1. Halt the script by entering **<Ctrl-C>**.

2. Kill any DCE daemons by executing **dce.clean**, which is also installed in **/etc**.

3. Remove all database files and directories that were created so far by executing **dce.rm**, also in **/etc**.

   The following files get deleted by this script:

   dce_cf.db
   var/rpc/rpcdep.dat
   var/rpc/rpcdllb.dat
   etc/security/pe_site
   var/security/.mkey*
   var/security/creds/*
   var/security/lrgy_lock
   var/security/rgy_data*
   /krb5/v5srvtab
   var/directory/cds/*
   var/directory/cds/adm/cdsd/*
   var/directory/cds/adm/gdad/*
   var/adm/directory/cds/*
   var/adm/directory/cds/cdsadv/cdsadv.log
   etc/cds_config
   etc/cds.conf
   etc/cds_defaults
   etc/gda_id
   etc/cdsadv.pid
   etc/cdscache.shmid
   etc/cdsd.pid
   etc/gdsconfig
   var/directory/gds/dsa/dir*
   var/directory/gds/cache/*
   var/adm/directory/gds/adm/*
   var/adm/directory/gds/conf/*

var/adm/time/mgt_acl
var/adm/time/dts_shared_memory_id

The **dce.rm** command does not currently remove the file system database and directories.

In addition, you must clean out the DCE registry information before retrying to configure a node. You must enter the following lines to the **rgy_edit** command:

domain principal
delete hosts/*hostname***/self**
delete hosts/*hostname***/cds-server**
delete hosts/*hostname***/gda**

If you merely halt and restart the script, it will fail if it encounters existing files in the databases that it tries to build.

Chapter 8

Phase 1: Initial Cell Configuration

This chapter walks you through the creation and configuration of the first Security Server and the first CDS Server. It also explains the entries that are required in the Security and CDS databases. Additional server configurations are detailed in Chapter 9, ''Phase 2: Configuring DCE Clients and Additional DCE Services.''

8.1 Preparation for Configuration

Before configuring any server or client, you must install the binaries for that configuration. The **dce_config** installation and configuration script can perform these functions:

- The script can install the required binaries for the type of server that is specified, creating the necessary local directory structures. The script uses the **/opt/dcelocal** (**$DCELOCAL**) directory as the default directory for all DCE data.
- The script can also install client binaries on the server.

- The script can create the DCE configuration file **/opt/dcelocal/dce_cf.db** with read, write, and execute permissions only by **root**, and read and execute permissions for other users. This file contains the following information:

  — The hostname from which the host principal name is derived

  — The cell name

The installation procedures are described in Chapter 7 of this guide. When you have installed all of the server and client binaries, you can start configuring the DCE servers.

You must first configure the Security Server and a CDS Server. These servers can be configured on the same or on different machines. If they are on different machines, your cell has a ''split server'' configuration, and you must bring up both servers before bringing up either client. Consult the *OSF DCE Release Notes* for the current information on timing and sequence constraints when bringing up split server cells.

You must ensure that the Security Server and the server or client that you are configuring have clocks that are at least loosely synchronized. If these clocks differ by a significant amount (approximately 5 minutes or more), the configuration will fail.

Some of the following sections provide verification steps to help you gauge the progress that was made in the configuration steps that you have just completed.

Both the Security Server and the CDS Server maintain databases. Sufficient disk space must be available on the machine(s) that are to be configured as these servers. You need enough space for the two databases if a single machine is to operate as both servers.

8.2 Security Server Configuration

The Security Server must be the first machine that is configured in the cell because every other DCE server must authenticate itself with the Security Service. Be sure to install the Security Server binaries on this machine first.

To configure the Security Server, the script does the following:

1. The script creates **/opt/dcelocal/dce_cf.db**.

2. It removes any Security configuration files that are in the **/opt/dcelocal/var/security**, **/opt/dcelocal/etc/security**, **/krb5**, **/tmp**, and **/usr/tmp** directories. This step ensures a clean configuration of the Security Server.

   Note: Step 3 is optional. It can be omitted if interoperability with MIT Kerberos applications is not required.

3. The script configures the system for interoperability with MIT Kerberos applications with the following steps:

   a. The script creates **/krb5/krb.conf**, which is in the following format:

      cellname
      cellname   hostname

      where *cellname* is the name of the cell that is being configured, and *hostname* is the name of the machine that is being configured.

   b. It adds the following entry to **/etc/services**:

      ```
      kerberos    88/udp          kdc
      ```

      The registered Kerberos IP port is 88. If an existing entry in **/etc/services** is found with a different port number, you are asked if it should be changed. Unless you have a reason to do otherwise, you should change it to 88.

4. The script starts the **rpcd** daemon, which allows the Security Server to register its endpoint.

5. The script initializes the Security Service by using the **sec_create_db** command to create the database for the Security registry.

6. The script starts the **secd** daemon.

   The following message appears when **secd** starts because the DCE Security Service is attempting to register with a directory service. This error warns the user that **secd** is unable to find a directory server with which to register. When the initial CDS Server is configured, **secd** registers with CDS and this message disappears.

```
SECD Incomplete NSI registration; task sleeping for 60 seconds
```

7. The script starts the **sec_clientd** daemon, which periodically refreshes the ticket-granting ticket for the machine's principal. The **sec_clientd** daemon also validates the Security Server that the **dce_login** command contacts.

8. The script authenticates the user as the cell administrator by using **-dce-** as the default password.

   The script prompts the administrator to use **rgy_edit** to change the password when the user exits the script. Note that if you change the cell administrator password, you must also change the value of the *cellpw* variable in **dce_config** accordingly in order for the configuration script to continue to work correctly.

9. The script adds the following groups to the registry database by using the **rgy_edit** command:

   none
   system
   daemon
   uucp
   bin
   kmem
   mail
   tty
   tcb
   subsys/dce/sec-admin
   subsys/dce/cds-admin
   subsys/dce/dfs-admin
   subsys/dce/dts-admin
   subsys/dce/dskl-admin
   subsys/dce/dts-servers
   subsys/dce/cds-server
   subsys/dce/dfs-fs-servers

subsys/dce/dfs-bak-servers
acct-admin

10. The script adds the following principals to the registry database:

    hosts/*hostname***/self**
    krbtgt/.../*cellname*
    dce-ptgt
    dce-rgy

11. The script adds the following accounts to the registry database; all accounts are created with a default password of **-dce-**:

    nobody
    root
    daemon
    uucp
    sys
    who
    mail
    tbc
    bin
    dce-ptgt
    dce-rgy
    cell_admin
    krbtgt/.../*cellname*
    hosts/*hostname***/self**

12. The script adds the following organization to the registry database:

    none

13. The script assigns the cell administrator to the following groups:

    acct-admin
    subsys/dce/sec-admin
    subsys/dce/cds-admin
    subsys/dce/dts-admin
    subsys/dce/dfs-admin
    subsys/dce/dskl-admin

14. The script ensures that the **hosts/***hostname***/self** machine principal is a member of the **subsys/dce/dts-servers** group.

15. The script creates the **keytab** entries for the **/hosts/***hostname***/self** principal.

Verification Steps:

1. At this point a **dce_login cell-admin -dce-** command succeeds on the machine configured as the Security Server.
2. Also, you can use **rgy_edit** to view the Security database.

8.3 Initial CDS Server Configuration

You can configure the CDS Server on the same machine as the Security Server, or on a different machine. The CDS Server must be the second server that is configured in the cell because other servers require specific entries in the CDS namespace before they are started up.

Once the CDS Server and DCE Client binaries are installed on the machine, and you select `Initial Cell Configuration` (Option 1) from the `DCE Configuration Menu` (see Figure 7-4), the script does the following:

1. The script requests the name of the cell, without the leading **/.../** prefix.
2. The script creates **/opt/dcelocal/dce_cf.db** and sets its permission bits to 644, if this machine is not also the Security Server machine.

   If this machine is the Security Server, **dce_cf.db** was created when the Security Server was configured.
3. The script starts **rpcd**, if the daemon is not already running.

   Note: The following step is done only if the CDS Server machine is different from the Security Server machine.
4. The script configures this machine as a Security client.

   Note: Thc following step happens whether or not the machine is the same as the Security Server.
5. The script creates the CDS configuration file **/opt/dcelocal/etc/cds.conf**. This step allows CDS to use the principal names now known to the DCE Security Service. The file is formatted as follows:

   cds.cdsd.security.server_princ_name: hosts/*hostname***/cds-server**
   cds.gdad.security.server_princ_name: hosts/*hostname***/gda**

cds.*.security.host_princ_name: hosts/*hostname*/self
cds.*.security.server_group_name: subsys/dce/cds-server
cds.*.security.admin_group_name: subsys/dce/cds-admin

where *hostname* is the name of the current machine.

The *hostname* that is specified in **cds.conf** must match the value in **dce_cf.db**. The script ensures that these values match.

Now the CDS daemon processes can be started. The script does the following:

1. The script starts the **cdsadv** daemon. This daemon starts a **cdsclerk** daemon when the first clerk request is made.
2. The script starts the **cdsd** daemon by using the **-a** flag. This step initializes the CDS namespace (see Section 7.4).
3. The script starts the **gdad** daemon.

   Before configuring a GDA Server for DNS, the machine you use must be able to access the **named** daemon from a file such as **resolv.conf**. However, **named** is not necessary for configuring GDA with only the **X500** option enabled.

   For further information about GDA configuration, see Chapter 9.
4. The script authenticates the user as the cell administrator. This step is necessary if the *cell_administrative_user* is not a member of the **subsys/dce/cds-admin** group when the CDS configuration routine is invoked.

Verification Steps:

1. If you execute **dce_login** and then **klist**, the information about your Security context is displayed. If you do not execute **dce_login**, then **klist** only returns

```
/tmp/dcecred_ffffffff
```

2. At this point, the command

   cdscp show dir /.:

   provides a listing of information about the CDS root directory's attributes.

Note: There needs to be a clearinghouse name that is equivalent to */.../cellname/hostname*_**ch**. Also, the replica type is **master**.

8.4 Initializing the Cell Namespace

The script performs all of the following steps, which are necessary to complete the initial configuration of the CDS namespace. These steps do not require interaction with the administrator.

1. In the CDS namespace, the following directories and objects are created by using the **cdscp** program:

   /.:/cell-profile
   /.:/lan-profile
   /.:/subsys
   /.:/subsys/dce
   /.:/subsys/dce/sec
   /.:/subsys/dce/dfs
   /.:/subsys/dce/dfs/bak
   /.:/hosts
   /.:/hosts/*hostname*
   /.:/sec
   /.:/sec-v1
   /.:/hosts/*hostname***/self**
   /.:/hosts/*hostname***/profile**

2. Then, the **rpccp** program is used to associate the objects that are listed, as follows:

   a. The **/.:/sec** object becomes an element of **/.:/cell-profile**.

   b. An interface UUID is associated with **/.:/cell-profile** for **secidmap**, **krb5rpc**, and **rpriv**. These are the interface names that are used by the applications to reach the DCE Security Service.

c. The **/.:/lan-profile** object becomes an element of **/.:/cell-profile**.

   If your cell uses more than one LAN, you must configure a separate LAN profile for each LAN with which the cell connects.

d. Bindings are exported for the CDS Server and clerk ACL Manager UUIDs.

e. An entry is created for **/...**/*cellname*/**fs**.

f. An entry is created for **/.:/subsys/dce/dfs/bak**.

g. The **/...**/*cellname*/**fs** and **/.:/subsys/dce/dfs/bak** entries are members of **/...**/*cellname*/**hosts**/*hostname*.

h. The ACLs are set on each directory, object, or entry.

Verification Steps:

1. At this point, the following command succeeds:

   cdscp show dir *dirname*

   where *dirname* is a directory such as **/.:/subsys/dce**. This command provides information that is similar to that given in the CDS Server verification steps in the previous section.

2. You can also use **rpccp** and other **cdscp** commands to verify the cell configuration.

8.5 Multiple LAN Profiles

If your cell is using multiple Local Area Networks, you may want to use different LAN profiles to organize how applications find objects in the namespace. For example, the DTS global servers require that different LAN profiles be used for machines that reside in different LANs. Otherwise, they will not act as global servers and will use all the other **dtsd** servers for synchronization.

When configuring a CDS Server or a CDS client, the script will ask you the following questions:

```
Is this cell using multiple LANs?
Specifically, are clients and servers divided into profile
groups to facilitate performance?  Many cells will not require
this feature.

Are you using multiple LANs within this cell? (n)
```

In most configurations, you can take the default answer, **n**. However, if your cell uses multiple LANs, answer **y** and enter the local LAN name when you are prompted for it. The script will create a LAN profile in the namespace. The information in the LAN profile is used by DTS.

Chapter 9

Phase 2: Configuring DCE Clients and Additional DCE Services

This chapter describes how to configure additional DCE servers and a DCE Client once **rpcd**, the Security Service, and a CDS Server are configured, and the cell namespace is initialized. Use the DCE installation and configuration script **dce_config** to continue the configuration process.

When you select an option from the `Additional Server Configuration` menu, you must be a member of the **subsys/dce/cds-admin**, **subsys/dce/sec-admin**, and **acct-admin** groups. This criterion is satisfied by running the script while authenticated as the *cell_administrative_user*.

All references to functions that are provided by the DCE installation and configuration script refer to menus and submenus that you will see while running the script.

To configure GDS, refer to the *OSF DCE Administration Guide—Extended Services*. GDS provides a separate menu-driven interface to help you with configuring, activating, and initializing the Global Directory Service.

9.1 Configuring a DCE Client

This section describes the steps that are followed by the script to configure a DCE Client. Each machine must be configured as a DCE Client before it is configured as a server of any type.

If you want to change a machine from a DCE Client configuration to a DCE server configuration, be sure to stop the appropriate client daemons before doing so. For example, if you want to configure a DCE Client machine as a DTS Server, stop the **dtsd** daemon.

You must ensure that the Security Server and the DCE Client that you are configuring have clocks that are at least loosely synchronized. If these clocks differ by a significant amount (approximately 5 minutes or more), the configuration will fail.

When you select the DCE Client configuration, the script does the following:

1. The script checks that you have installed the DCE Client binaries on this machine by verifying that **libdce.a** is installed.
2. The script checks whether this machine is a Security Server. If it is not, the script configures the machine as a Security client.
3. The script authenticates the user as the cell administrator with the following command:

   dce_login *cell_administrative_user password*

4. The script creates **/opt/dcelocal/dce_cf.db**, if it does not already exist on this machine.
5. The script starts the **rpcd** daemon.
6. The script configures this machine as a Security client if the **secd** daemon is not running by completing the following steps:
   a. The script obtains the name of the Security Server.
   b. The script adds the following entry to **/etc/services**:

      ```
      kerberos    88/udp   kdc
      ```

The registered Kerberos IP port is 88. If an existing entry in **/etc/services** is found with a different port number, you are asked if it should be changed. Unless you have a reason to do otherwise, you should change it to 88.

c. The script creates the **/krb5/krb.conf** and *dcelocal***/etc/security/pe_site** files to match the Security Server's version of these files.

d. Using the **rgy_edit** command, the script creates a keytab entry for this client machine's principal with the following command:

ktadd -p hosts/*hostname***/self**

e. The script starts the **sec_clientd** daemon.

7. The script configures this machine as a CDS client if **cdsd** is not running by completing the following steps:

a. The script creates the **/opt/dcelocal/cds.conf** file.

b. The script starts the **cdsadv** process and establishes contact with the CDS Server.

c. The script authenticates with the **dce_login** command to verify that the user is a member of the **cds-admin** group.

d. The script prompts the user for CDS Server and local host information.

Note: If you are configuring a DCE Client machine into a cell that is using a WAN, you must give the full DNS machine names; for example, **machineA@osf.org**, not simply **machineA**. In addition, you must answer **no** to the question: `Can` *local host* `broadcast to` *CDS server machine*`?`

8. The script creates the following entries in the namespace:

a. Directory **/.:/hosts/***hostname*

b. Objects

   /.:/hosts/*hostname***/self**

   /.:/hosts/*hostname***/cds-clerk**

   /.:/hosts/*hostname***/profile**

9. The script configures this machine as a DTS clerk if **dtsd** is not running by completing the following steps:

   a. The script starts the **dtsd** daemon.

   b. The script executes the following commands:

```
dtscp create type clerk
dtscp enable
```

10. The script prompts the user to indicate if this machine needs to be configured as a DFS client. The script configures a DFS client as follows:

    a. On all DFS client machines with extensible kernels that are not also DFS Servers, it loads DFS into the kernel with the following commands:

```
/opt/dcelocal/ext/cfgexport -a /opt/dcelocal/ext/export.ext
/opt/dcelocal/ext/cfgdfs -a /opt/dcelocal/ext/dfscore.ext
/opt/dcelocal/ext/cfgdfs -a /opt/dcelocal/ext/dfscmfx.ext
```

Note: These commands are specific to the RS/6000 reference platform that is running AIX. Because DFS Servers can also be DFS clients, DFS may have been loaded into the kernel in the earlier server step. In this case, omit this step.

On systems without extensible kernels, such as the OSF/1 reference platform, the DFS modules link in at boot time, so the script does not do anything to the kernel at this point. Therefore, the actions taken in this step for the RS/6000 are a proper superset of what happens on an OSF/1 system. However, the steps that follow happen in common for both types of platforms.

b. The script starts the **dfsbind** portion of the Cache Manager as follows:

   dcelocal/**bin/dfsbind&**

c. The script prompts the user to select whether the cache is in memory or on the local disk before creating **CacheInfo**.

d. If the cache is on disk, the script starts the **dfsd** daemon as follows:

   dfsd &

e. If the cache is in memory, the script starts the **dfsd** daemon by using the appropriate flags, as follows:

   dfsd -memcache -blocks *size*

   where *size* is the size of the cache.

11. The script creates *dcelocal*/**etc/CacheInfo** after prompting you for the following:

    a. The global root

    b. The cache size (the number of 1K blocks; the default is 10,000)

    c. The name of the cache directory

    Typically, you should accept the defaults by entering a carriage return.

9.2 Configuring an Additional CDS Server

The following subsections describe how to configure an additional CDS Server using the configuration script.

9.2.1 Prerequisites

Before an additional CDS Server can be configured, you must complete the initial cell configuration that is described in Chapter 8. Also, this machine must not be the same machine on which the initial CDS Server was configured.

9.2.2 Configuration Steps

This section describes how to configure an additional CDS Server after the initial CDS Server in your cell is configured.

When you select `Additional CDS Server(s)` from the `Additional Server Configuration` menu, the script does the following:

1. The script obtains the name of the initial CDS Server and verifies that this is not the same machine.
2. It verifies that the **cdsd** daemon has been installed and starts it.
3. The script starts the **rpcd** daemon if it is not already running.
4. It configures this machine as a Security client if it is not a Security Server.
5. The script issues the **dce_login** *cell_administrative_user password* command, where *cell_administrative_user* is the cell administrator that is created by using the **rgy_edit** command during the initial cell configuration.
6. It obtains the name of the new clearinghouse that is to be created on this machine, and verifies that it is not the same as the clearinghouse on the initial CDS Server.
7. The script creates a new clearinghouse on this CDS Server and creates a replica of the root directory.
8. It replicates any other directories that you specify.
9. The script configures (or does not configure) this machine as a GDA Server, according to the option that is selected from the GDA configuration submenu.

9.3 Configuring an Additional Security Server

The following subsections describe how to configure an additional Security Server using the configuration script.

9.3.1 Prerequisites

Before an additional Security Server can be configured, you must complete the initial cell configuration that is described in Chapter 8. Also, this machine must not be the same machine on which a Security Server has already been configured.

9.3.2 Configuration Steps

This section describes how to configure an additional Security Server after the initial Security Server and CDS Server in your cell have been configured.

When you select `Replica Security server` from the `Additional Server Configuration` menu, the script performs the following steps:

1. The script obtains the name of the new Security replica.
2. It starts **rpcd** if it is not already running.
3. The script configures this machine as a Security client.
4. It then configures this machine as a CDS client.
5. The script modifies the ACLs on the Security replica list so that this host may become a Security replica.
6. It modifies the ACLs on the CDS Security directory and the Security group so that this host may add its name to the Security directory and the Security group.
7. The script executes **sec_create_db** to create a stub Security database. This database will be initialized by the master later.

8. It starts the Security Server on this machine. This new server's database will be initialized by the master or another up-to-date replica.

9.4 Configuring DTS

The following subsections describe how the script configures a DTS local server, a DTS global server, or a DTS clerk. Before you configure DTS, you must complete the initial cell configuration that is described in Chapter 8.

The configuration script does not configure a time provider. (See the *OSF DCE Administration Guide—Core Components* for information about the time providers; see also the *OSF DCE Administration Reference* for additional information about the TPI for DTS software.)

9.4.1 Prerequisites

The following requirements, which were completed during the initial cell configuration process, must be complete before you can configure a DTS local or global server, or a DTS clerk. The script verifies that the following conditions are met:

- The user has the authority to create entries in the Security and CDS namespaces when appropriate.
- The **/.:/cell-profile** and **/.:/lan-profile** objects exist. The **/.:/cell-profile** object contains an entry for the LAN-Services UUID.
- This machine has access to the CDS and Security namespaces as a CDS and Security client, and the **rpcd** daemon is running.
- The **/opt/dcelocal/dce_cf.db** file exists on this machine and contains entries for the name of the cell and the name of the local host.
- A **hosts/***hostname***/self** entry exists in the **subsys/dce/dts-servers** group for this machine's principal.
- A **/.:/hosts/***hostname***/profile** object exists on this machine and contains a default entry that points to the cell profile.

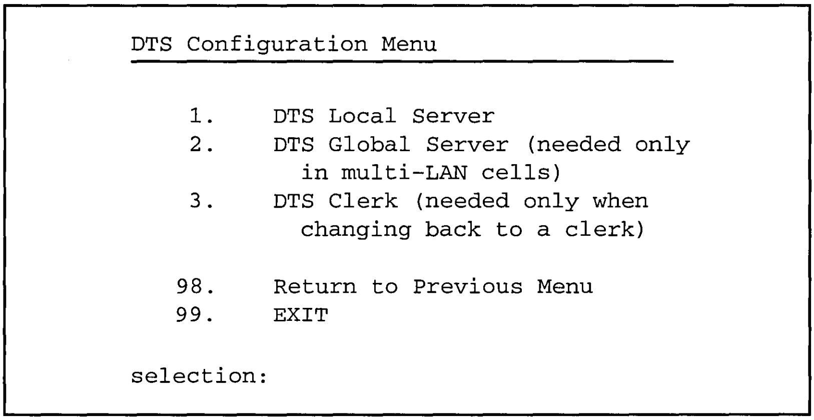
- The following principal and account entries exist in the registry:

DTS Servers Group	**subsys/dce/dts-servers**
DTS Administration Group	**subsys/dce/dts-admin**

9.4.2 Configuration Steps

When you select `DTS` from the `Additional Server Configuration` menu, the script presents the submenu that is shown in Figure 9-1.

Figure 9–1. DTS Configuration Menu



```
DTS Configuration Menu
_______________________________________

    1.     DTS Local Server
    2.     DTS Global Server (needed only
             in multi-LAN cells)
    3.     DTS Clerk (needed only when
             changing back to a clerk)

   98.     Return to Previous Menu
   99.     EXIT

selection:
```

The script then executes the following steps, some of which depend on your selection:

1. The script starts up the DTS daemon, **dtsd**. The **dtsd** daemon must access the Security Server at initialization time by using the machine's principal login context, **hosts/***hostname***/self**.

   Before starting up DTS, the script verifies that authentication has occurred as a member of the cell administrator's group.

2. If you select the `DTS Local Server` option, the script creates and enables DTS by using the control program, **dtscp**, and executing the following **dtscp** commands:

   create type server
   set courier role noncourier
   enable

3. If you select the `DTS Global Server` option, the script executes the following **dtscp** commands:

   create type server
   set courier role noncourier
   enable
   advertise

4. If you select the `DTS Clerk` option, the script creates a DTS clerk by using the following **dtscp** commands:

   create type clerk
   enable

5. If you select option `98`, the script returns to the `Additional Server Configuration` menu. If you select option `99`, the script exits. All other responses are rejected.

The *OSF DCE Administration Guide—Core Components* describes how you can reconfigure DTS interactively, if this becomes necessary, after the initial DCE configuration is complete.

9.5 Configuring GDA

The following subsections describe how to configure GDA in a cell that uses GDS or DNS for Global Directory Service. Configuration of GDA for DNS needs to be done manually.

9.5.1 Prerequisites

GDA configuration is part of the CDS Server configuration process.

If DNS will be used, then the machine that you configure as the GDA Server must have access to the **named** daemon through **resolv.conf**.

The script verifies that the following conditions are met:

- This machine is configured as a CDS and Security client and the **rpcd** daemon is running.
- The **/opt/dcelocal/dce_cf.db** file exists on this machine and contains entries for the name of the cell and the name of the local host.
- The principal and account entries exist in the registry for **hosts/***hostname***/gda** and a **ktab** entry exists for this principal.

9.5.2 Configuration Steps

To configure GDA, the script starts the **gdad** daemon. No further configuration is required for a DCE configuration that uses GDS.

To configure GDA for a DCE configuration that uses DNS, you must also manually complete the following steps:

1. Obtain the output of the **cdscp show cell** command and then register this information about your cell with the **named** daemon, which is the server that controls DNS name resolution.
2. Restart the **named** daemon.

The **gdad** daemon can be run in the following modes:

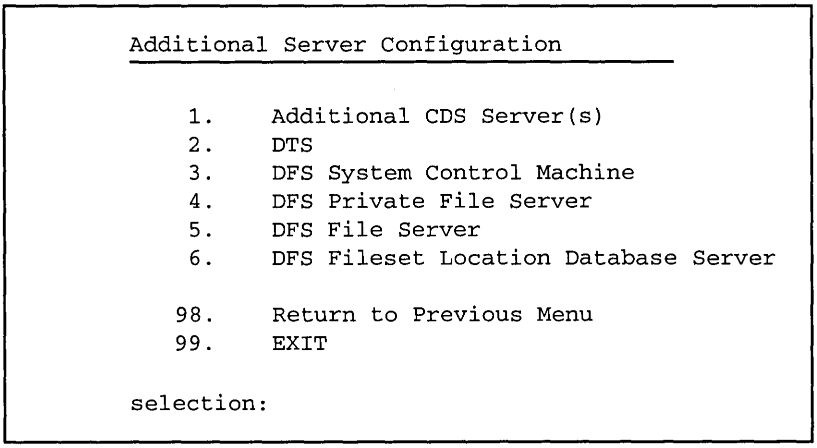
DNS	**gdad -x**
X.500	**gdad -b**
Both DNS and X.500	**gdad**

(See the *OSF DCE Administration Guide—Core Components* for information about starting and stopping **gdad**, and for how to use the output of **cdscp show cell**.)

9.6 Configuring DFS

The following subsections describe the steps that are involved in configuring DFS. The script configures DFS from the menu that is shown in Figure 9-2.

Figure 9–2. Additional Server Configuration Menu



```
Additional Server Configuration

        1.     Additional CDS Server(s)
        2.     DTS
        3.     DFS System Control Machine
        4.     DFS Private File Server
        5.     DFS File Server
        6.     DFS Fileset Location Database Server

       98.     Return to Previous Menu
       99.     EXIT

selection:
```

Note: Before using the script to configure DFS, refer to the *OSF DCE Release Notes* for the current DFS configuration information and the configuration steps, which may supersede some of the steps in this description.

9.6.1 Prerequisites

Before beginning the DFS configuration process, be sure that you have made the planning decisions that are described in the *OSF DCE Administration Guide—Extended Services*.

These decisions include the following:

- Choosing the first DFS machine
- Determining the cache size on DFS client machines
- Defining DFS domains
- Setting up your cell's DFS tree structure
- Determining which file system to use for **root.dfs**
- Manually creating DCE LFS or non-LFS filesets before creating and mounting **root.dfs**

9.6.1.1 Creating the root.dfs Fileset

You need to create the file system for the **root.dfs** fileset. This file system can be either a DCE LFS fileset or a non-LFS file system (non-LFS fileset). To use a non-LFS file system, create the file system manually by using your operating system's file system commands. To use a DCE LFS fileset, create a DCE LFS aggregate with the DFS **newaggr** command.

(See the *OSF DCE Administration Guide—Extended Services* before setting up your **root.dfs** fileset.)

9.6.1.2 Preliminary Steps

The following requirements, which were completed during the initial cell configuration process, are verified by the script before it allows you to configure a DFS Server:

- The user has the authority to create entries in the Security and CDS namespaces when appropriate.
- This machine has access to the CDS and Security namespaces as a CDS and Security client and the **rpcd** daemon is running.
- The **/opt/dcelocal/dce_cf.db** file exists on this machine and contains entries for the name of the cell and the name of the local host.

- A **hosts/***hostname***/self** entry exists in the **subsys/dce/dfs-fs-servers** group for this machine's principal.
- The following principal and account entries exist in the registry:

DFS Server Group	**subsys/dce/dfs-fs-servers**
DFS Administration Group	**subsys/dce/dfs-admin**

You must also ensure that you have the correct kernel installed on your system. For the DECstation 3100 reference port, copy either **efsvmunix** or **dfsvmunix** from the install tree to **/vmunix** on the system and then reboot it. The **efsvmunix** OSF/1 kernel includes support for both DFS and DCE LFS. The **dfsvmunix** OSF/1 kernel includes support only for DFS but requires less disk and memory resources. Save a copy of your existing OSF/1 kernel until you have tested the newly installed kernel.

9.6.2 Configuration Steps

The following subsections describe the steps that are required to make the following DFS configurations:

- DFS System Control machine
- DFS Fileset Location Database Server
- DFS Private File Server
- DFS File Server

Use the script to accomplish the DFS configuration tasks that are described in the following subsections.

The script assumes that the Fileset Location Database Server is on the machine that exports **root.dfs**.

9.6.2.1 Configuring a DFS System Control Machine

The script requires that you configure the DFS System Control machine first because it creates the administrative lists for the various DFS services (**flserv**, **ftserv**, **up**, and **bak**). When you do so, the script does the following:

1. It loads kernel extensions on systems with dynamically extensible kernels and initializes the entries for Security and CDS.
2. It starts the **bosserver** if it is not already running.
3. It adds the minimum set of information to the BOS Server's administrative list with the following command:

   ```
   bos admin -server /.:/hosts/$HOSTNAME -adminlist admin.bos \
           -group $DFS_ADMIN_GROUP -noauth -createlist
   ```

4. It activates the authorization checks.
5. It creates the following administrative lists in the **/opt/dcelocal/bin/bos** directory:

   admin.bak
   admin.fl
   admin.ft
   admin.up

6. It starts the **upserver**, exporting the preceding administrative lists.

9.6.2.2 Configuring a DFS Fileset Location Database Server

Each cell that uses DFS requires at least one FLDB Server. Configure this server immediately after configuring the DFS System Control machine. To do so, select option 6 from the `Additional Server Configuration` menu.

The **root.dfs** fileset that is to be exported must be created manually by using your operating system's normal file system commands. If the **root.dfs** fileset is on any file system other than DCE LFS, that file system must be mounted. The aggregate name is the directory mount point.

When you specify the aggregate name in response to the prompt, you must specify it as its mounted file system name, not as its physical device name. For example, if you want to use the device **/dev/rz3c**, which is mounted as **/var**, you should specify **/var** as the aggregate name. Ignore the following message that is displayed by the script:

```
Aggregate Id 1 does not exist on the server
```

When configuring a DFS Fileset Location Database Server, the script does the following:

1. On a UNIX system with a dynamically extensible kernel, such as the RS/6000 AIX reference port, the script configures DFS into the kernel with the following commands:

   ```
   /opt/dcelocal/ext/cfgexport -a /opt/dcelocal/ext/export.ext
   /opt/dcelocal/ext/cfgdfs -a /opt/dcelocal/ext/dfscore.ext
   /opt/dcelocal/ext/cfgdfs -a /opt/dcelocal/ext/dfscmfx.ext
   ```

   Note: The preceding commands are RS/6000 specific. In traditional UNIX systems (without extensible kernels), DFS is configured into the kernel, and the DFS libraries are linked when the system is booted. Once started, DFS-related daemons cannot be killed without rebooting the machine.

2. At this point, on systems with extensible kernels, the DCE LFS kernel extension is also loaded into the kernel. However, the following command is optional if DCE LFS is not part of your configuration:

   ```
   /opt/dcelocal/ext/cfgdfs -a /opt/dcelocal/ext/dcelfs.ext
   ```

   Again, for nonextensible kernels, the DCE LFS libraries are linked in at boot time.

The script then does the following:

1. It creates a Security group for this workstation, or updates an existing group by using **dfs_rgyinit**. This step involves adding the DFS Server hostname to the group.

2. It verifies that the file system that is used for **root.dfs** is mounted.

3. It adds principals and passwords to the Security registry for this **flserver** host.
4. It starts the **bosserver** and configures it to control **flserver**, **ftserver**, and **repserver**.
5. It prompts you for the name of the DFS System Control machine. When you enter that name, the script strips off any IP domain information, if necessary, before checking that the DFS System Control machine and the **fldb** machine are different.
6. It creates **dfstab** by entering one set of values if the **root.dfs** fileset is a non-LFS file system and another set of values if the **root.dfs** fileset is a DCE LFS file system.
7. It executes **dfsexport** as follows:

   dcelocal/**bin/dfsexport -a**

8. It configures this machine as a DFS client.
9. It starts the **fxd** daemon.

9.6.2.3 Configuring a DFS Private File Server

When configuring a DFS Private File Server, the script does the following:

1. First, on machines that are running AIX, which has an extensible kernel, the script configures DFS into the kernel with the following commands:

   ```
   /opt/dcelocal/ext/cfgexport -a /opt/dcelocal/ext/export.ext
   /opt/dcelocal/ext/cfgdfs -a /opt/dcelocal/ext/dfscore.ext
   /opt/dcelocal/ext/cfgdfs -a /opt/dcelocal/ext/dfscmfx.ext
   ```

   In traditional UNIX systems, and in particular the OSF/1 reference platform, DFS is configured into the kernel and is already loaded when the machine is booted.

2. On AIX systems, the DCE LFS kernel extension is also loaded into the kernel if the machine uses DCE LFS. The following command is superfluous if DCE LFS is not part of your configuration:

   ```
   /opt/dcelocal/ext/cfgdfs -a /opt/dcelocal/ext/dcelfs.ext
   ```

   Again, on machines that are running OSF/1 Release 1.1.1, the local file system is configured into the kernel at boot time.

The script then does the following:

1. It adds principals and passwords to the registry for this **ftserver** host.

Sample Principal	**hosts/***hostname*/**dfs-server**
Sample Account	**hosts***hostname*/**dfs-server**

2. It authenticates the user as a member of the cell administrator's group, as shown in the following example:

   ```
   dce_login cell_administrative_user password
   ```

3. It starts the **bosserver** and configures it to control the **ftserver**.

4. It configures the DFS Private File Server machine as a DFS client if the machine is not already a DFS client. (See Section 9.1 for more information on this process.)

5. It starts the **fxd** process with the following command:

   ```
   /opt/dcelocal/bin/fxd -mainprocs dfs_threads -admin group group_name
   ```

   where the default value for *dfs_threads* is **7**.

9.6.2.4 Configuring a DFS File Server

Configuration of a DFS File Server is the same as configuration of a DFS Private File Server, except that the File Server runs the **repserver** daemon. The script does the following:

1. First, DFS is configured into the kernel with the following commands:

   ```
   /opt/dcelocal/ext/cfgexport -a /opt/dcelocal/ext/export.ext
   /opt/dcelocal/ext/cfgdfs -a /opt/dcelocal/ext/dfscore.ext
   /opt/dcelocal/ext/cfgdfs -a /opt/dcelocal/ext/dfscmfx.ext
   ```

   Note: The preceding commands are RS/6000 specific. In traditional UNIX systems, DFS is configured into the kernel and is already loaded when the kernel is booted.

   The DCE LFS kernel extension is also loaded into the kernel. The following command is optional if DCE LFS is not part of your configuration:

   ```
   /opt/dcelocal/ext/cfgdfs -a /opt/dcelocal/ext/dcelfs.ext
   ```

2. It adds principals and passwords to the registry for each **flserver** and **ftserver** host.

3. It creates a Security account for this machine's administrator.

4. It authenticates the user as a member of the cell administrator's group, as shown in the following example:

   ```
   dce_login cell_administrative_user password
   ```

5. It starts **bosserver** and configures it to control **ftserver** and **repserver**.

6. It starts **ftserver** with the following command:

   ```
   bos create -server /.:/hosts/hostname -process ftserver \
     -type simple -cmd /opt/dcelocal/bin/ftserver
   ```

7. It starts the replication server with the following command:

   ```
   bos create -server /.:/hosts/hostname -process repserver \
             -type simple -cmd /opt/dcelocal/bin/repserver
   ```

8. The File Server machine is then configured as a DFS client. (See Section 9.1 for more information on this process.)

9. The script then starts **fxd** with the following command:

   ```
   /opt/dcelocal/bin/fxd -mainprocs dfs_threads -admin group group_name
   ```

   where the default value for *dfs_threads* is **7**.

Appendix A

The DCE Cell Namespace

This appendix describes the names that CDS and the DCE Security Service use within the DCE cell namespace. Most of these namespace entries are created during the initial DCE configuration.

In the tables that follow, the ''CDS Class'' field is either used internally by the **CDS_Clearinghouse** entry or by the CDS Browser. The ''Well Known'' field specifies whether the last component of a name is an architecturally required name. The ''Initial Configuration ACL'' field specifies the ACL that is created by running the DCE configuration script. The ''Created By'' field specifies how this entry is created.

The *hostname*, *lclhostname*, *cellname*, and *creator* entries are defined as follows:

- *hostname*

  This is a cell-relative hostname. For example, the *hostname* for a host named **machine1.abc.com** is **machine1**. Note that for cells with subdomains, a directory structure is possible. For example, the host **apollo.mercury.acs.cmu.edu** can have a *hostname* of **acs/mercury/apollo**.

- *lclhostname*

  This is the single component hostname. This name is always the least significant component of the hostname. The *lclhostname* for the examples given previously are **abc** and **apollo**.

- *cellname*

  This is the global name of the cell, without the special character string **/.../**; for example, **seattle.abc.com** or **C=US/O=ABC/OU=Seattle**.

- *creator*

  This is the name of the principal that created the cell.

A.1 The CDS Space

Figures A-1 through A-3 illustrate the CDS namespace within the DCE cell namespace. The subsections that follow provide a description of each entry.

Figure A–1. The Top-Level CDS Directory

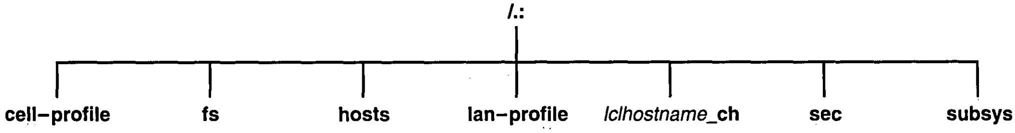


Figure A–2. The CDS hosts Directory

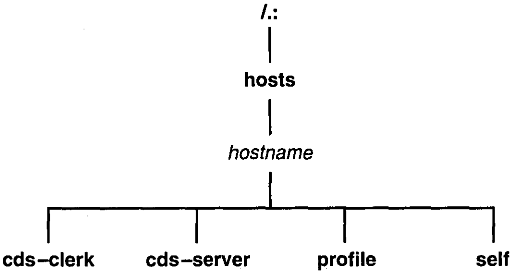
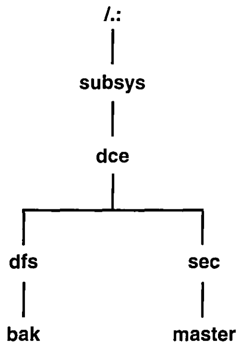


Figure A–3. The CDS subsys Directory



A.1.1 The Top-Level CDS Directory

The following tables describe the namespace entries for **/.:**, which is the top-level CDS directory.

Name	*/.:*
CDS Type	Directory
Well Known	Yes
Description	This is the cell root directory. The special character string **/.:** is a shorthand form of */.../cellname*. This directory is replicated in every clearinghouse.
Initial Configuration ACL	
Object ACL	**Default cell = */.../cellname*** **unauthenticated:r--t---** **group:subsys/dce/cds-admin:rwdtcia** **group:subsys/dce/cds-server:rwdtcia** **any_other:r--t---**

Name	**/.:**
Default Object ACL	**Default cell =** /.../*cellname* **unauthenticated:r--t---** **group:subsys/dce/cds-admin:rwdtc--** **group:subsys/dce/cds-server:rwdtc--** **any_other:r--t---**
Default Container ACL	**Default cell =** /.../*cellname* **unauthenticated:r--t---** **group:subsys/dce/cds-admin:rwdtcia** **group:subsys/dce/cds-server:rwdtcia** **any_other:r--t---**
Created By	CDS configuration

Name	**/.:/cell-profile**
CDS Type	Object
CDS Class	**RPC_Profile**
Well Known	Yes
Description	This is the master default profile for the cell. Ultimately, all host, user, and other profiles must link to this profile. This profile is created at cell creation and must include the following entry: **LAN-Services-UUID** /.../*cellname*/**lan-profile** Note that like all profile entries, only global names can be used. This profile must include interfaces for the Privilege Server, the Registry Server, and the Authentication Server. In multi-LAN cells, this is the profile in which the DTS global set entries are entered.

Name	**/.:/cell-profile**
Initial Configuration ACL	
Object ACL	**unauthenticated:r--t-** **user:**_creator_**:rwdtc** **group:subsys/dce/cds-admin:rwdtc** **group:subsys/dce/cds-server:rwdtc** **group:subsys/dce/dts-admin:rw-t-** **group:subsys/dce/dts-servers:rw-t-** **any_other:r--t-**
Created By	DCE configuration

Name	**/.:/fs**
CDS Type	Object
CDS Class	**RPC_Group**
Well Known	No
Description	The RPC bindings of all Fileset Database machines that house the FLDB are listed in this group. This group consists of RPC bindings of the following form: _/.../cellname/_**hosts/**_hostname/_**self** This object must have a single object UUID attached to it. This is the junction to the DFS filespace within the cell namespace. The character string **/:** is a CDS soft link to **/.:/fs**.
Initial Configuration ACL	
Object ACL	**Default cell =** _/.../cellname_ **unauthenticated:r--t-** **user:**_creator_**:rwdtc** **group:subsys/dce/cds-admin:rwdtc** **group:subsys/dce/cds-server:rwdtc** **group:subsys/dce/dfs-fs-servers:rwdtc** **group:subsys/dce/dfs-admin:rwdtc** **any_other:r--t-**
Created By	DCE configuration

Name	**/.:/hosts**
CDS Type	Directory
Well Known	No
Description	The host directories are cataloged here.
Initial Configuration ACL	
Object ACL	**Default cell =** */.../cellname* **unauthenticated:r--t---** **user:*creator*:rwdtcia** **user:hosts/*hostname*/cds-server:rwdtcia** **user:hosts/*hostname*/self:rwdtcia** **group:subsys/dce/cds-admin:rwdtcia** **group:subsys/dce/cds-server:rwdtcia** **any_other:r--t---**
Default Object ACL	**Default cell =** */.../cellname* **unauthenticated:r--t---** **group:subsys/dce/cds-admin:rwdtc--** **group:subsys/dce/cds-server:rwdtc--** **any_other:r--t---**
Default Container ACL	**Default cell =** */.../cellname* **unauthenticated:r--t---** **group:subsys/dce/cds-admin:rwdtcia** **group:subsys/dce/cds-server:rwdtcia** **any_other:r--t---**
Created By	DCE configuration

Name	**/.:/lan-profile**
CDS Type	Object
CDS Class	**RPC_Profile**
Well Known	No
Description	This is the default LAN profile that is used by DTS, and potentially by other services. In single LAN cells, this is the profile in which entries for the DTS local set entries are entered.
Initial Configuration ACL	
Object ACL	**unauthenticated:r--t-** **user:***creator***:rwdtc** **group:subsys/dce/cds-admin:rwdtc** **group:subsys/dce/cds-server:rwdtc** **group:subsys/dce/dts-admin:rwdtc** **group:subsys/dce/dts-servers:rwdtc** **any_other:r--t-**
Created By	DCE configuration

Name	**/.:/***lclhostname***_ch**
CDS Type	Object
CDS Class	**CDS_Clearinghouse**
Well Known	No
Description	All clearinghouses are cataloged in the cell root. This name is only fixed for the first CDS Server that you configure. You can choose different names for any additional CDS Servers that you configure.
Initial Configuration ACL	
Object ACL	**unauthenticated:r--t-** **group:subsys/dce/cds-admin:rwdtc** **group:subsys/dce/cds-server:rwdtc** **any_other:r--t-**
Created By	CDS configuration

Name	***/.:/sec***
CDS Type	Object
CDS Class	**RPC_Group**
Well Known	No
Description	This is the RPC group of all Security Servers for this cell. It contains the entries */.../cellname/***subsys/dce/sec/master** and */.../cellname/***subsys/dce/sec/rep_1**. This is the junction into the Security namespace.
Initial Configuration ACL	
Object ACL	**Default cell =** */.../cellname* **unauthenticated:r--t-** **user:***creator***:rwdtc** **user:dce-rgy:rwdtc** **user:hosts/***rep_1_hostname***/self:rwdtc** **group:subsys/dce/cds-admin:rwdtc** **group:subsys/dce/cds-server:rwdtc** **group:subsys/dce/sec-admin:rwdtc** **any_other:r--t-**
Created By	DCE configuration

Name	**/.:/subsys**
CDS Type	Directory
Well Known	No
Description	This directory contains directories for different subsystems in this cell. It contains the **dce** subdirectory. It is recommended that companies adding subsystems to DCE conform to the convention of creating a unique directory below **subsys** by using their trademark as a directory name (**/.:/subsys/***trademark*). These directories are used for storage of location-independent information about services. Server entries, groups, and profiles for the entire cell should be stored in directories below **subsys**.
Initial Configuration ACL	
Object ACL	**Default cell =** */.../cellname* **unauthenticated:r--t---** **user:***creator***:rwdtcia** **user:hosts/***hostname***:rwdtcia** **group:subsys/dce/cds-admin:rwdtcia** **group:subsys/dce/cds-server:rwdtcia** **any_other:r--t---**
Default Object ACL	**Default cell =** */.../cellname* **unauthenticated:r--t---** **group:subsys/dce/cds-admin:rwdtc--** **group:subsys/dce/cds-server:rwdtc--** **any_other:r--t---**
Default Container ACL	**Default cell =** */.../cellname* **unauthenticated:r--t---** **group:subsys/dce/cds-admin:rwdtcia** **group:subsys/dce/cds-server:rwdtcia** **any_other:r--t---**
Created By	DCE configuration

A.1.2 The CDS hosts Directory

The following tables describe the namespace entries for **/.:/hosts**, which is the CDS **hosts** directory.

Name	**/.:/hosts/*hostname***
CDS Type	Directory
Well Known	No
Description	Each host has a directory in which RPC server entries, groups, and profiles that are associated with this host are stored. This is simply a CDS directory. No bindings are present in the directory object itself; entries exist beneath the directory.
Initial Configuration ACL	
Object ACL	**Default cell = /.../*cellname*** **unauthenticated:r--t---** **user:*creator:*rwdtcia** **user:hosts/*hostname*/cds-server:rwdtcia** **user:hosts/*hostname*/self:rwdtcia** **group:subsys/dce/cds-admin:rwdtcia** **group:subsys/dce/cds-server:rwdtcia** **any_other:r--t---**
Default Object ACL	**Default cell = /.../*cellname*** **unauthenticated:r--t---** **group:subsys/dce/cds-admin:rwdtc--** **group:subsys/dce/cds-server:rwdtc--** **any_other:r--t---**
Default Container ACL	**Default cell = /.../*cellname*** **unauthenticated:r--t---** **group:subsys/dce/cds-admin:rwdtcia** **group:subsys/dce/cds-server:rwdtcia** **any_other:r--t---**
Created By	DCE configuration

Name	**/.:/hosts/*hostname*/cds-clerk**
CDS Type	Object
CDS Class	**RPC_Entry**
Well Known	No
Description	This entry contains the binding for a CDS clerk. This entry is used by the ACL editor to manage the ACL interface.
Initial Configuration ACL	
Object ACL	**Default cell =** */.../cellname* **unauthenticated:r--t-** **user:***creator***:rwdtc** **user:hosts/***hostname***/self:rw-t-** **group:subsys/dce/cds-admin:rwdtc** **group:subsys/dce/cds-server:rwdtc** **any_other:r--t-**
Created By	DCE configuration

Name	**/.:/hosts/*hostname*/cds-server**
CDS Type	Object
CDS Class	**RPC_Entry**
Well Known	No
Description	This entry contains the binding for a CDS Server. This entry is used by the ACL editor to manage the ACL interface.
Initial Configuration ACL	
Object ACL	**Default cell =** */.../cellname* **unauthenticated:r--t-** **user:***creator***:rwdtc** **user:hosts/***hostname***/self:rw-t-** **group:subsys/dce/cds-admin:rwdtc** **group:subsys/dce/cds-server:rwdtc** **any_other:r--t-**
Created By	DCE configuration

Name	**/.:/hosts/**_hostname_**/profile**
CDS Type	Object
CDS Class	**RPC_Entry**
Well Known	No
Description	This is the default profile for host *hostname*. It must contain a default that points (possibly indirectly) at **/.:/cell-profile**. Programs obtain this name by using the call **dce_cf_profile_name_from_host( )**.
Initial Configuration ACL	
Object ACL	**Default cell =** /.../*cellname* **unauthenticated:r--t-** **user:**_creator_**:rwdtc** **user:hosts/**_hostname_**/self:rw-t-** **group:subsys/dce/cds-admin:rwcdt** **group:subsys/dce/cds-server:rwcdt** **any_other:r--t-**
Created By	DCE configuration

Name	**/.:/hosts/**_hostname_**/self**
CDS Type	Object
CDS Class	**RPC_Entry**
Well Known	Yes
Description	This entry contains a binding to the **rpcd** daemon on host *hostname*. The **dce_cf_binding_entry_from_host( )** call returns either the name of this entry when provided a hostname, or the current host when a hostname is not provided.

Name	**/.:/hosts/*hostname*/self**
Initial Configuration ACL	
Object ACL	**Default cell** = /.../*cellname* **unauthenticated:r--t-** **user:*creator*:rwdtc** **user:hosts/*hostname*/self:rwrdtc** **group:subsys/dce/cds-admin:rwdtc** **group:subsys/dce/cds-server:rwdtc** **any_other:r--t-**
Created By	DCE configuration

A.1.3 The CDS subsys Directory

The following tables describe the namespace entries for **/.:/subsys**, which is the CDS **subsys** directory.

Name	**/.:/subsys/dce**
CDS Type	Directory
Well Known	No
Description	This directory contains DCE-specific names.
Initial Configuration ACL	
Object ACL	**Default cell** = /.../*cellname* **unauthenticated:r--t---** **user:*creator*:rwdtcia** **user:hosts/*hostname*/cds-server:rwdtcia** **group:subsys/dce/cds-admin:rwdtcia** **group:subsys/dce/cds-server:rwdtcia** **any_other:r--t---**
Default Object ACL	**Default cell** = /.../*cellname* **unauthenticated:r--t---** **group:subsys/dce/cds-admin:rwdtc--** **group:subsys/dce/cds-server:rwdtc--** **any_other:r--t---**

Name	**/.:/subsys/dce**
Default Container ACL	**Default cell** = */.../cellname* **unauthenticated:r--t---** **group:subsys/dce/cds-admin:rwdtcia** **group:subsys/dce/cds-server:rwdtcia** **any_other:r--t---**
Created By	DCE configuration

Name	**/.:/subsys/dce/dfs**
CDS Type	Directory
Well Known	No
Description	This directory contains all of the DFS-specific names.
Initial Configuration ACL	
Object ACL	**Default cell** = */.../cellname* **unauthenticated:r--t---** **user:***creator***:rwdtcia** **user:hosts/***hostname***/cds-server:rwdtcia** **group:subsys/dce/cds-admin:rwdtcia** **group:subsys/dce/cds-server:rwdtcia** **group:subsys/dce/dfs-admin:rwdtcia** **any_other:r--t---**
Default Object ACL	**Default cell** = */.../cellname* **unauthenticated:r--t---** **group:subsys/dce/cds-admin:rwdtc--** **group:subsys/dce/cds-server:rwdtc--** **group:subsys/dce/dfs-admin:rwdtc--** **any_other:r--t---**
Default Container ACL	**Default cell** = */.../cellname* **unauthenticated:r--t---** **group:subsys/dce/cds-admin:rwdtcia** **group:subsys/dce/cds-server:rwdtcia** **group:subsys/dce/dfs-admin:rwdtcia** **any_other:r--t---**
Created By	DCE configuration

Name	**/.:/subsys/dce/dfs/bak**
CDS Type	Object
CDS Class	**RPC_Entry**
Well Known	No
Description	The RPC bindings of all Backup Database machines that are storing the Backup Database are listed in this entry. This entry is similar to the **/.:/fs** group in that its members are RPC bindings of the */.../cellname*/**hosts**/*hostname*/**self** form. In addition, this group must have a single object UUID attached to it.
Initial Configuration ACL	
Object ACL	**Default cell =** */.../cellname* **unauthenticated:r--t-** **user:***creator***:rwdtc** **user:hosts/***hostname***/cds-server:rwdtc** **group:subsys/dce/cds-admin:rwdtc** **group:subsys/dce/cds-server:rwdtc** **any_other:r--t-**
Created By	DCE configuration

Name	**/.:/subsys/dce/sec**
CDS Type	Directory
Well Known	No
Description	This directory contains Security-specific names.
Initial Configuration ACL	
Object ACL	**Default cell =** */.../cellname* **unauthenticated:r--t---** **user:***creator***:rwdtcia** **user:hosts/***hostname***/cds-server:rwdtcia** **user:dce-rgy:rwdtci-** **user:hosts/***rep_1_hostname***/self:rwdtia** **group:subsys/dce/cds-admin:rwdtcia** **group:subsys/dce/cds-server:rwdtcia** **group:subsys/dce/sec-admin:rwdtcia** **any_other:r--t---**

Name	**/.:/subsys/dce/sec**
Default Object ACL	__Default cell = /.../__*cellname* **unauthenticated:r--t---** **user:dce-rgy:rwdt---** **user:hosts/***rep_1_hostname***/self:rwdtc** **group:subsys/dce/cds-admin:rwdtc--** **group:subsys/dce/cds-server:rwdtc--** **group:subsys/dce/sec-admin:rwdtc--** **any_other:r--t---**
Default Container ACL	__Default cell = /.../__*cellname* **unauthenticated:r--t---** **user:dce-rgy:rwdtci-** **user:hosts/***rep_1_hostname***/self:rwdtcia** **group:subsys/dce/cds-admin:rwdtcia** **group:subsys/dce/cds-server:rwdtcia** **group:subsys/dce/sec-admin:rwdtcia** **any_other:r--t---**
Created By	DCE configuration

Name	**/.:/subsys/dce/sec/master**
CDS Type	Object
CDS Class	**RPC_Entry**
Well Known	No
Description	This is the server entry for the master Security Server for this cell. The bindings for the Registry Server, the Privilege Server, and the Authentication Server are exported by the Registry Server to this entry.

Name	**/.:/subsys/dce/sec/master**
Initial Configuration ACL	
Object ACL	**Default cell =** */.../cellname* **unauthenticated:r--t-** **user:dce-rgy:rwdt-** **user:**_creator_**:rwdtc** **group:subsys/dce/cds-admin:rwdtc** **group:subsys/dce/cds-server:rwdtc** **group:subsys/dce/sec-admin:rwdtc** **any_other:r--t-**
Created By	DCE configuration

Name	**/.:/subsys/dce/sec/rep_1**
CDS Type	Object
CDS Class	**RPC_Entry**
Well Known	No
Description	This is the server entry for a slave Security Server for this cell. The bindings for the Registry Server, the Privilege Server, and the Authentication Server are exported by the Registry Server to this entry.
Initial Configuration ACL	
Object ACL	**Default cell =** */.../cellname* **unauthenticated:r--t-** **user:dce-rgy:rwdt-** **user:**_creator_**:rwdtc** **user:hosts/**_rep_1_hostname_**/self:rwdtc** **group:subsys/dce/cds-admin:rwdtc** **group:subsys/dce/cds-server:rwdtc** **group:subsys/dce/sec-admin:rwdtc** **any_other:r--t-**
Created By	DCE configuration

A.2 The Security Space

Figures A-4 through A-6 illustrate the Security namespace within the DCE cell namespace. The subsections that follow provide a description of each entry. The subdirectories that comprise the Security namespace are **principal**, **group**, **org**, and **policy**.

To view the ACLs on any of these namespace entries, you need to include the name of the Security junction. For example, the group name **acct-admin** is referenced as **/.:/sec/group/acct-admin** when you use the **acl_edit** command.

In contrast to the **acl_edit** command, the **rgy_edit** command operates on a principal, group, or organization name without including **/.:/sec** and **principal**, **group**, or **org** as part of the name. To operate on the group **acct-admin** using **rgy_edit**, you specify the domain **group** and the group name **acct-admin**.

Figure A–4. The Top-Level Security Directory

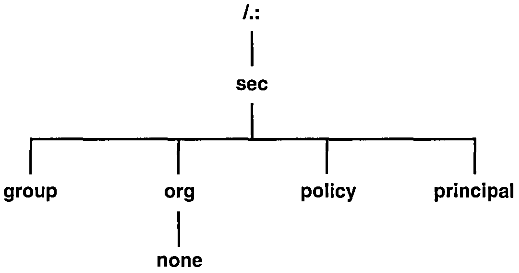


Figure A–5. The sec/group Directory

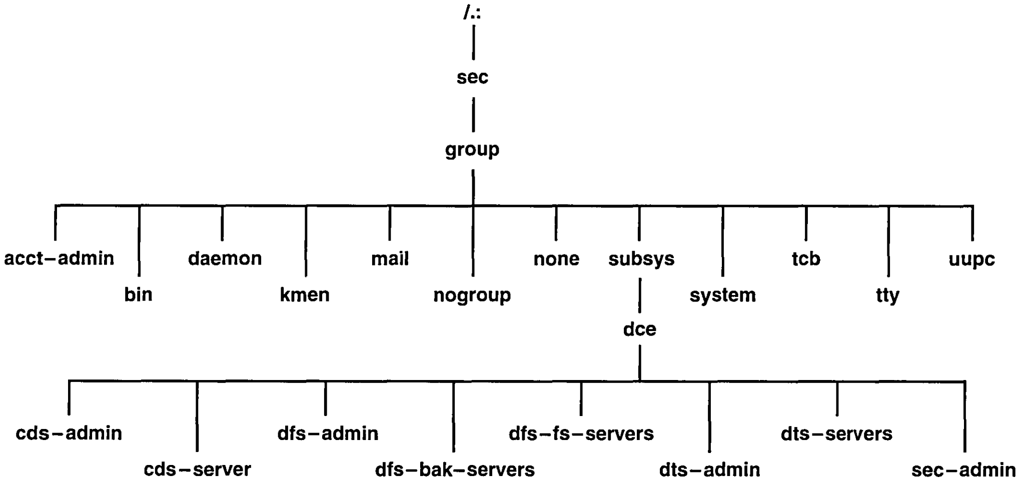
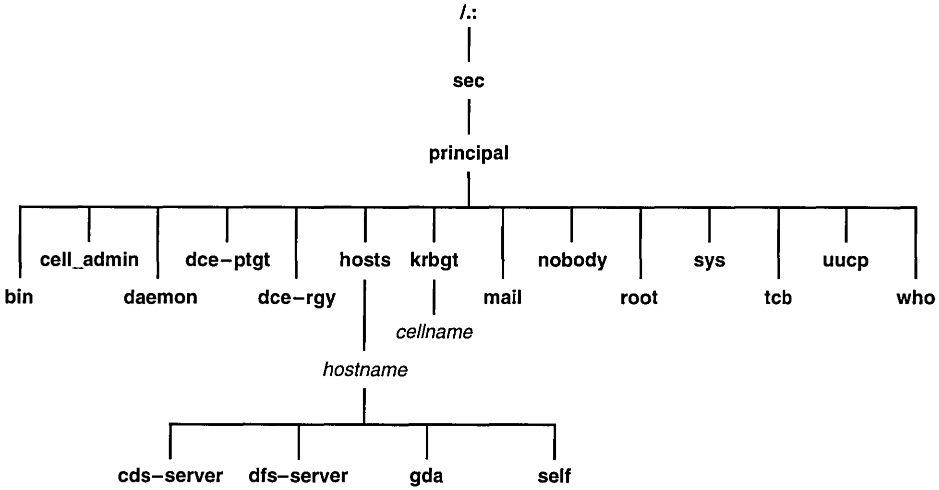


Figure A–6. The sec/principal Directory



In the following subsections, descriptions of entries in an initial Security namespace are given. Included is the suggested UNIX user identifier (UNIX UID) or group identifier (UNIX GID) that they are assigned to. Vendors should use these values if possible; however, be aware that they may change in future revisions. Some entries are assigned the next available identifier, starting with 100; therefore, these may vary from cell to cell. They are indicated as ''Generated''.

A.2.1 The Top-Level Security Directory

The following tables describe the namespace entries for **/.:/sec**, which is the top-level Security directory.

Name	**/.:/sec/group**
Well Known	Yes. This name is not architecturally defined, but is defined by the implementation.
Description	This is the Security directory that holds all of the groups. This name is only used by the ACL editor.
Initial Configuration ACL	
Object ACL	**Default cell =** *I.../cellname* **unauthenticated:r-----** **user:***creator***:rcidDn** **group:acct-admin:rcidDn** **other_obj:r-----** **any_other:r-----**
Default Object ACL	**Default cell =** *I.../cellname* **unauthenticated:r-t-----** **user:***creator***:rctDnfmM** **group_obj:r-t-----** **group:acct-admin:rctDnfmM** **other_obj:r-t-----** **any_other:r-------**

Name	**/.:/sec/group**
Default Container ACL	**Default cell = /.../*cellname*** **unauthenticated:r-----** **user:***creator***:rcidDn** **group:acct-admin:rcidDn** **other_obj:r-----** **any_other:r-----**
Created By	Security configuration

Name	**/.:/sec/org**
Well Known	Yes. This name is not architecturally defined, but is defined by the implementation.
Description	This is the Security directory that holds all of the organizations. This name is only used by the ACL editor.
Initial Configuration ACL	
Object ACL	**Default cell = /.../*cellname*** **unauthenticated:r-----** **user:***creator***:rcidDn** **group:acct-admin:rcidDn** **other_obj:r-----** **any_other:r-----**
Default Object ACL	**Default cell = /.../*cellname*** **unauthenticated:r-t-----** **user:***creator***:rctDnfmM** **group:acct-admin:rctDnfmM** **other_obj:r-t-----** **any_other:r-t-----**

Name	**/.:/sec/org**
Default Container ACL	**Default cell =** */.../cellname* **unauthenticated:r-----** **user:***creator***:rcidDn** **group:acct-admin:rcidDn** **other_obj:r-----** **any_other:r-----**
Created By	DCE configuration

Name	**/.:/sec/org/none**
Well Known	Yes
Description	This is the default organization.
Initial Configuration ACL	
Object ACL	**Default cell =** */.../cellname* **unauthenticated:r-t-----** **user:***creator***:rctDnfmM** **group:acct-admin:rctDnfmM** **other_obj:r-t-----** **any_other:r-t-----**
UNIX Org ID	12
Created By	Security configuration

Name	**/.:/sec/policy**
Well Known	Yes. This name is not architecturally defined, but is defined by the implementation.
Description	This entry's ACL controls the ability to set Security policies on a cell-wide basis.
Initial Configuration ACL	
Object ACL	**Default cell =** */.../cellname* **unauthenticated:r----** **user:**_creator_**:rcmaA** **group:acct-admin:rcmaA** **other_obj:r----** **any_other:r----**
Created By	DCE configuration

Name	**/.:/sec/principal**
Well Known	Yes. This name is not architecturally defined, but it cannot be changed in DCE V1.0.
Description	This is the Security directory that holds all of the principals. This name is only used by the ACL editor.
Initial Configuration ACL	
Object ACL	**Default cell =** */.../cellname* **unauthenticated:r-----** **user:**_creator_**:rcidDn** **group:acct-admin:rcidDn** **other_obj:r-----** **any_other_obj:r-----**
Default Object ACL	**Default cell =** */.../cellname* **unauthenticated:r-------g** **user_obj:r---f--ug** **user:**_creator_**:rcDnfmaug** **group:acct-admin:rcDnfmaug** **other_obj:r-------g** **any_other:r--------**

Name	**/.:/sec/principal**
Default Container ACL	**Default cell = /.../*cellname*** **unauthenticated:r-----** **user:***creator***:rcidDn** **group:acct-admin:rcidDn** **other_obj:r-----** **any_other:r-----**
Created By	Security configuration

A.2.2 The sec/group Directory

The following tables describe the namespace entries for **/.:/sec/group**, which is the Security **sec/group** directory.

Name	**/.:/sec/group/acct-admin**
Security Type	Group
Well Known	No
Description	This is the only group of principals that can create accounts.
Initial Configuration ACL	
Object ACL	**Default cell = /.../*cellname*** **unauthenticated:r-t-----** **user:***creator***:rctDnfmM** **group_obj:rctDnfmM** **other_obj:r-t-----** **any_other:r-t-----**
UNIX GID	Generated
Created By	DCE configuration

Name	**/.:/sec/group/bin**
Well Known	No
Description	This is the group for system binaries.
Initial Configuration ACL	
Object ACL	**Default cell =** */.../cellname*
	unauthenticated:r-t-----
	user:*creator***:rctDnfmM**
	group_obj:r-t-----
	group:acct-admin:rctDnfmM
	other_obj:r-t-----
	any_other:r-t-----
UNIX GID	3
Created By	Security configuration

Name	**/.:/sec/group/daemon**
Well Known	No
Description	This is the group for daemons.
Initial Configuration ACL	
Object ACL	**Default cell =** */.../cellname*
	unauthenticated:r-t-----
	user:*creator***:rctDnfmM**
	group_obj:r-t-----
	group:acct-admin:rctDnfmM
	other_obj:r-t-----
	any_other:r-t-----
UNIX GID	1
Created By	Security configuration

Name	**/.:/sec/group/kmem**
Well Known	No
Description	This is the group that has read access to kernel memory.
Initial Configuration ACL	
Object ACL	**Default cell =** */.../cellname*
	unauthenticated:r-t-----
	user:*creator***:rctDnfmM**
	group_obj:r-t-----
	group:acct-admin:rctDnfmM
	other_obj:r-t-----
	any_other:r-t-----
UNIX GID	4
Created By	Security configuration

Name	**/.:/sec/group/mail**
Well Known	No
Description	This is the group for the mail subsystem.
Initial Configuration ACL	
Object ACL	**Default cell =** */.../cellname*
	unauthenticated:r-t-----
	user:*creator***:rctDnfmM**
	group_obj:r-t-----
	group:acct-admin:rctDnfmM
	other_obj:r-t-----
	any_other:r-t-----
UNIX GID	6
Created By	Security configuration

Name	**/.:/sec/group/nogroup**
Well Known	Yes
Description	This is the default group for NFS access; it goes with user ID **nobody**.
Initial Configuration ACL	
Object ACL	**Default cell = /.../*cellname*** **unauthenticated:r-t-----** **user:***creator***:rctDnfmM** **group_obj:r-t-----** **group:acct-admin:rctDnfmM** **other_obj:r-t-----** **any_other:r-t-----**
UNIX GID	-2
Created By	Security configuration

Name	**/.:/sec/group/none**
Well Known	Yes
Description	This member does not belong to a group; it is the default group.
Initial Configuration ACL	
Object ACL	**Default cell = /.../*cellname*** **unauthenticated:r-t-----** **user:***creator***:rctDnfmM** **group_obj:r-t-----** **group:acct-admin:rctDnfmM** **other_obj:r-t-----** **any_other:r-t-----**
UNIX GID	12
Created By	Security configuration

Name	**/.:/sec/group/subsys**
Security Type	Group Directory
Well Known	Yes
Description	This directory contains **dce**. (See **/.:/subsys** in the CDS namespace.)
Initial Configuration ACL	
Object ACL	**Default cell = /.../***cellname* **unauthenticated:r-----** **user:***creator***:rcidDn** **group:acct-admin:rcidDn** **other_obj:r-----** **any_other:r-----**
Default Object ACL	**Default cell = /.../***cellname* **unauthenticated:r-t-----** **user:creator:rctDnfmM** **group_obj:r-t-----** **group:acct-admin:rctDnfmM** **other_obj:r-t-----** **any_other:r-t-----**
Default Container ACL	**Default cell = /.../***cellname* **unauthenticated:r-----** **user:creator:rcidDn** **group:acct-admin:rcidDn** **other_obj:r-----** **any_other:r-----**
Created By	DCE configuration

Name	**/.:/sec/group/system**
Well Known	No
Description	This is the group for system accounts.
Initial Configuration ACL	
Object ACL	**Default cell = /.../***cellname* **unauthenticated:r-t-----** **user:***creator***:rctDnfmM** **group_obj:r-t-----** **group:acct-admin:rctDnfmM** **other_obj:r-t-----** **any_other:r-t-----**
UNIX GID	0
Created By	Security configuration

Name	**/.:/sec/group/tcb**
Well Known	No
Description	This is the group that is used by security policy daemons on OSF/1 C2/B1 secure systems.
Initial Configuration ACL	
Object ACL	**Default cell = /.../***cellname* **unauthenticated:r-t-----** **user:***creator***:rctDnfmM** **group_obj:r-t-----** **group:acct-admin:rctDnfmM** **other_obj:r-t-----** **any_other:r-t-----**
UNIX GID	18
Created By	Security configuration

Name	**/.:/sec/group/tty**
Well Known	No
Description	This is the group that has write access to terminals.
Initial Configuration ACL	
Object ACL	**Default cell = /.../***cellname* **unauthenticated:r-t-----** **user:***creator***:rctDnfmM** **group_obj:r-t-----** **group:acct-admin:rctDnfmM** **other_obj:r-t-----** **any_other:r-t-----**
UNIX GID	7
Created By	Security configuration

Name	**/.:/sec/group/uucp**
Well Known	No
Description	This is the group for the UUCP subsystem.
Initial Configuration ACL	
Object ACL	**Default cell = /.../***cellname* **unauthenticated:r-t-----** **user:***creator***:rctDnfmM** **group_obj:r-t-----** **group:acct-admin:rctDnfmM** **other_obj:r-t-----** **any_other:r-t-----**
UNIX GID	2
Created By	Security configuration

A.2.3 The sec/group/subsys Directory

The following tables describe the namespace entries for **/.:/sec/group/subsys**, which is the Security **sec/group/subsys** directory.

Name	**/.:/sec/group/subsys/dce**
Security Type	Group Directory
Well Known	Yes
Description	This directory contains the DCE required groups.
Initial Configuration ACL	
Object ACL	**Default cell =** */.../cellname* **unauthenticated:r-----** **user:**_creator_**:rcidDn** **group:acct-admin:rcidDn** **other_obj:r-----** **any_other:r-----**
Default Object ACL	**Default cell =** */.../cellname* **unauthenticated:r-t-----** **user:**_creator_**:rctDnfmM** **group_obj:r-rt-----** **group:acct-admin:rcitDnfmM** **other_obj:r-t-----** **any_other:r-t-----**
Default Container ACL	**Default cell =** */.../cellname* **unauthenticated:r-----** **user:**_creator_**:rcidDn** **group:acct-admin:rcidDn** **other_obj:r-----** **any_other:r-----**
Created By	DCE configuration

Name	**/.:/sec/group/subsys/dce/cds-admin**
Security Type	Group
Well Known	No
Description	This is the administrative group that is on the default ACLs for administrative objects. Clearinghouses have this group on their ACLs with all rights. The first user of the cell must be added to this group immediately after creation.
Initial Configuration ACL	
Object ACL	**Default cell =** */.../cellname* **unauthenticated:r-t-----** **user:***creator***:rctDnfmM** **group_obj:r-t-----** **group:acct-admin:rctDnfmM** **other_obj:r-t-----** **any_other:r-t-----**
UNIX GID	Generated
Created By	DCE configuration

Name	**/.:/sec/group/subsys/dce/cds-server**
Security Type	Group
Well Known	Yes
Description	This is the group of all the CDS Servers for the local cell. As each new server is added to the cell, it must be added to this group. CDS Server authentication consists of checking for the server's membership in this group.

Name	**/.:/sec/group/subsys/dce/cds-server**
Initial Configuration ACL	
Object ACL	**Default cell =** */.../cellname* **unauthenticated:r-t-----** **user:***creator***:rctDnfmM** **group_obj:r-t-----** **group:acct-admin:rctDnfmM** **group:subsys/dce/cds-admin:rctDnfmM** **group:subsys/dce/cds-server:rctDnfmM** **other_obj:r-t-----** **any_other:r-t-----**
UNIX GID	Generated
Created By	DCE configuration

Name	**/.:/sec/group/subsys/dce/dfs-admin**
Security Type	Group
Well Known	No
Description	This is the DFS administrator's group. Members of this group have full permissions to alter the DFS configuration within the cell.
Initial Configuration ACL	
Object ACL	**Default cell =** */.../cellname* **unauthenticated:r-t-----** **user:***creator***:rctDnfmM** **group_obj:r-t-----** **group:acct-admin:rctDnfmM** **other_obj:r-t-----** **any_other:r-t-----**
UNIX GID	Generated
Created By	DCE configuration

Name	**/.:/sec/group/subsys/dce/dfs-bak-servers**
Security Type	Group
Well Known	Yes
Description	This is the Security group to which all Backup Database Servers belong. A server entry in the CDS group **/.:/subsys/dce/fs** is checked for authorization to act as a Backup Database Server by determining whether it belongs to this Security group.
Initial Configuration ACL	
Object ACL	**Default cell = /.../**_cellname_ **unauthenticated:r-t-----** **user:**_creator_**:rctDnfmM** **group_obj:r-t-----** **group:acct-admin:rctDnfmM** **other_obj:r-t-----** **any_other:r-t-----**
UNIX GID	Generated
Created By	DCE configuration

Name	**/.:/sec/group/subsys/dce/dfs-fs-servers**
Security Type	Group
Well Known	Yes
Description	Abbreviated forms of the DFS Server principals of all the Fileset Database machines are listed in this group. The abbreviated form of a machine's DFS Server principal that is stored in the group is of the form **hosts/**_hostnamedfs-server_. A server entry that is obtained from the CDS group **/.:/fs** is checked for authorization to act as a Fileset Location Server by determining if it belongs to this group.

Name	**/.:/sec/group/subsys/dce/dfs-fs-servers**
Initial Configuration ACL	
Object ACL	**Default cell =** */.../cellname*
	unauthenticated:r-t-----
	user:*creator***:rctDnfmM**
	group_obj:r-t-----
	group:acct-admin:rctDnfmM
	group:subsys/dce/dfs-admin:rctDnfmM
	other_obj:r-t-----
	any_other:r-t-----
UNIX GID	Generated
Created By	DCE configuration

Name	**/.:/sec/group/subsys/dce/dskl-admin**
Security Type	Group
Well Known	No
Description	This is the Diskless Service administrator's group.
Initial Configuration ACL	
Object ACL	**Default cell =** */.../cellname*
	unauthenticated:r-t-----
	user:*creator***:rctDnfmM**
	group_obj:r-t-----
	group:acct-admin:rctDnfmM
	other_obj:r-t-----
	any_other:r-t-----
UNIX GID	Generated
Created By	DCE configuration

Name	**/.:/sec/group/subsys/dce/dts-admin**
Security Type	Group
Well Known	No
Description	This is the DTS administrator's group. Members of this group have full permissions to administer DTS by adding servers and so forth.

Name	/.:/sec/group/subsys/dce/dts-admin
Initial Configuration ACL	
Object ACL	**Default cell =** */.../cellname*
	unauthenticated:r-t-----
	user:*creator***:rctDnfmM**
	group_obj:r-t-----
	group:acct-admin:rctDnfmM
	other_obj:r-t-----
	any_other:r-t-----
UNIX GID	Generated
Created By	DCE configuration

Name	/.:/sec/group/subsys/dce/dts-servers
Security Type	Group
Well Known	Yes
Description	This is the group of DTS Servers.
Initial Configuration ACL	
Object ACL	**Default cell =** */.../cellname*
	unauthenticated:r-t-----
	user:*creator***:rctDnfmM**
	group_obj:r-t-----
	group:acct-admin:rctDnfmM
	group:subsys/dce/dts-admin:rctDnfmM
	other_obj:r-t-----
	any_other:r-t-----
UNIX GID	Generated
Created By	DCE configuration

Name	/.:/sec/group/subsys/dce/sec-admin
Security Type	Group
Well Known	No
Description	This is the Security administrator's group. Members of this group have full permissions to administer the Security database.

Name	**/.:/sec/group/subsys/dce/sec-admin**
Initial Configuration ACL	
Object ACL	**Default cell =** */.../cellname*
	unauthenticated:r-t-----
	user:*creator***:rctDnfmM**
	group_obj:r-t-----
	group:acct-admin:rctDnfmM
	other_obj:r-t-----
	any_other:r-t-----
UNIX GID	Generated
Created By	DCE configuration

A.2.4 The sec/principal Directory

The following tables describe the namespace entries for **/.:/sec/principal**, which is the Security **sec/principal** directory.

Name	**/.:/sec/principal/bin**
Well Known	No
Description	This is the owner of the system binaries.
Initial Configuration ACL	
Object ACL	**Default cell =** */.../cellname*
	unauthenticated:r--------
	user_obj:r---f--ug
	user:*creator***:rcDnfmaug**
	group:acct-admin:rcDnfmaug
	other_obj:r-------g
	any_other:r--------
UNIX UID	3
Created By	Security configuration

Name	**/.:/sec/principal/cell_admin**
Well Known	No
Description	This is the DCE cell administrator.
Initial Configuration ACL	
Object ACL	**Default cell =** */.../cellname*
	unauthenticated:r--------
	user_obj:rcDnfmaug
	user:*creator***:rcDnfmaug**
	group:acct-admin:rcDnfmaug
	other_obj:r-------g
	any_other:r--------
UNIX UID	Generated
Created By	Security configuration

Name	**/.:/sec/principal/daemon**
Well Known	No
Description	This is the user for the various daemons.
Initial Configuration ACL	
Object ACL	**Default cell =** */.../cellname*
	unauthenticated:r--------
	user_obj:r---f--ug
	user:*creator***:rcDnfmaug**
	group:acct-admin:rcDnfmaug
	other_obj:r-------g
	any_other:r--------
UNIX UID	1
Created By	Security configuration

Name	**/.:/sec/principal/dce-ptgt**
Security Type	Principal
Well Known	Yes
Description	This is the architecturally defined principal name of the Privilege Server.
Initial Configuration ACL	
Object ACL	**Default cell = /.../***cellname* **unauthenticated:r--------** **user_obj:r---f--ug** **user:***creator***:rcDnfmaug** **group:acct-admin:rcDnfmaug** **other_obj:r-------g** **any_other:r--------**
UNIX UID	20
Created By	Security configuration

Name	**/.:/sec/principal/dce-rgy**
Security Type	Principal
Well Known	Yes
Description	This is the architecturally defined principal name of the Registry Server.
Initial Configuration ACL	
Object ACL	**Default cell = /.../***cellname* **unauthenticated:r--------** **user_obj:r---f--ug** **user:***creator***:rcDnfmaug** **group:acct-admin:rcDnfmaug** **other_obj:r-------g** **any_other:r--------**
UNIX UID	21
Created By	Security configuration

Name	**/.:/sec/principal/hosts**
Security Type	Principal Directory
Well Known	No
Description	This directory contains the **.:/hosts/***hostname* directories.
Initial Configuration ACL	
Object ACL	**Default cell = /.../***cellname* **unauthenticated:r-----** **user:***creator***:rcidDn** **group:acct-admin:rcidDn** **other_obj:r-----** **any_other:r-----**
Default Object ACL	**Default cell = /.../***cellname* **unauthenticated:r--------** **user_obj:r---f--ug** **user:***creator***:rcDnfmaug** **group:acct-admin:rcDnfmaug** **other_obj:r-------g** **any_other:r--------**
Default Container ACL	**Default cell = /.../***cellname* **unauthenticated:r-----** **user:***creator***:rcidDn** **group:acct-admin:rcidDn** **other_obj:r-----** **any_other:r-----**
Created By	Security configuration

Name	**/.:/sec/principal/krbtgt** *(also known as* **/...***)*
Security Type	Principal Directory
Well Known	Yes
Description	This is the architecturally specified name of the Security namespace where foreign cell names are cataloged. All cells that this cell communicates with appear here.

Name	**/.:/sec/principal/krbtgt** *(also known as* ***/...****)*
Initial Configuration ACL	
Object ACL	**Default cell =** */.../cellname* **unauthenticated:r-----** **user:***creator***:rcidDn** **group:acct-admin:rcidDn** **other_obj:r-----** **any_other:r-----**
Default Object ACL	**Default cell =** */.../cellname* **unauthenticated:r--------** **user_obj:r---f--ug** **user:***creator***:rcDnfmaug** **group:acct-admin:rcDnfmaug** **other_obj:r-------g** **any_other:r--------**
Default Container ACL	**Default cell =** */.../cellname* **unauthenticated:r-----** **user:***creator***:rcidDn** **group:acct-admin:rcidDn** **other_obj:r-----** **any_other:r-----**
Created By	Security configuration

Name	**/.:/sec/principal/krbtgt/***cellname* *(also known as* ***/.:****)*
Security Type	Principal
Well Known	No
Description	This is the principal of the Authentication Server of the cell named ***/.../****cellname*. In the local cell, this is the principal for the **/.:** cell.

Name	**/.:/sec/principal/krbtgt/**_cellname_ *(also known as* **/.:***)*
Initial Configuration ACL	
Object ACL	**Default cell = /.../**_cellname_ **unauthenticated:r-------g** **user_obj:r---f--ug** **user:**_creator_**:rcDnfmaug** **group:acct-admin:rcDnfmaug** **other_obj:r-------g** **any_other:r--------**
Created By	Security configuration

Name	**/.:/sec/principal/mail**
Well Known	No
Description	This is the user for the mail subsystem.
Initial Configuration ACL	
Object ACL	**Default cell = /.../**_cellname_ **unauthenticated:r--------** **user_obj:r---f--ug** **user:**_creator_**:rcDnfmaug** **group:acct-admin:rcDnfmaug** **other_obj:r-------g** **any_other:r--------**
UNIX UID	6
Created By	Security configuration

Name	**/.:/sec/principal/nobody**
Well Known	No
Description	This is the default user for NFS accesss.
Initial Configuration ACL	
Object ACL	**Default cell = /.../***cellname* **unauthenticated:r--------** **user_obj:r---f--ug** **user:***creator***:rcDnfmaug** **group:acct-admin:rcDnfmaug** **other_obj:r-------g** **any_other:r--------**
UNIX UID	2
Created By	Security configuration

Name	**/.:/sec/principal/root**
Well Known	No
Description	This is the local operating system superuser.
Initial Configuration ACL	
Object ACL	**Default cell = /.../***cellname* **unauthenticated:r--------** **user_obj:r---f--ug** **user:***creator***:rcDnfmaug** **group:acct-admin:rcDnfmaug** **other_obj:r-------g** **any_other:r--------**
UNIX UID	0
Created By	Security configuration

Name	**/.:/sec/principal/sys**
Well Known	No
Description	This is a user who is permitted to read devices but is not a superuser.
Initial Configuration ACL	
Object ACL	**Default cell = /.../***cellname* **unauthenticated:r--------** **user_obj:r---f--ug** **user:***creator***:rcDnfmaug** **group:acct-admin:rcDnfmaug** **other_obj:r-------g** **any_other:r--------**
UNIX UID	2
Created By	Security configuration

Name	**/.:/sec/principal/tcb**
Well Known	No
Description	This is the user for security policy daemons on OSF/1 C2/B1 secure systems.
Initial Configuration ACL	
Object ACL	**Default cell = /.../***cellname* **unauthenticated:r--------** **user_obj:r---f--ug** **user:***creator***:rcDnfmaug** **group:acct-admin:rcDnfmaug** **other_obj:r-------g** **any_other:r--------**
UNIX UID	9
Created By	Security configuration

Name	**/.:/sec/principal/uucp**
Well Known	No
Description	This is the user for the UUCP subsystem.
Initial Configuration ACL	
Object ACL	**Default cell =** */.../cellname* **unauthenticated:r--------** **user_obj:r---f--ug** **user:***creator***:rcDnfmaug** **group:acct-admin:rcDnfmaug** **other_obj:r-------g** **any_other:r--------**
UNIX UID	4
Created By	Security configuration

Name	**/.:/sec/principal/who**
Well Known	No
Description	This is the user for remote **who** access.
Initial Configuration ACL	
Object ACL	**Default cell =** */.../cellname* **unauthenticated:r--------** **user_obj:r---f--ug** **user:***creator***:rcDnfmaug** **group:acct-admin:rcDnfmaug** **other_obj:r-------g** **any_other:r--------**
UNIX UID	5
Created By	Security configuration

A.2.5 The sec/principal/hosts Directory

The following tables describe the namespace entries for **/.:/sec/principal/hosts**, which is the Security **sec/principal/hosts** directory.

Name	**/.:/sec/principal/hosts/*hostname***
Security Type	Principal Directory
Well Known	No
Description	This directory contains Security principals for host *hostname*.
Initial Configuration ACL	
Object ACL	**Default cell =** */.../cellname* **unauthenticated:r-----** **user:*creator*:rcidDn** **group:acct-admin:rcidDn** **other_obj:r-----** **any_other:r-----**
Default Object ACL	**Default cell =** */.../cellname* **unauthenticated:r-------g** **user_obj:r---f--ug** **user:*creator*:rcDnfmaug** **group:acct-admin:rcDnfmaug** **other_obj:r-------g** **any_other:r--------**
Default Container ACL	**Default cell =** */.../cellname* **unauthenticated:r-----** **user:*creator*:rcidDn** **group:acct-admin:rcidDn** **other_obj:r-----** **any_other:r-----**
Created By	Security configuration

Name	**/.:/sec/principal/hosts/*hostname*/dts-entity**
Security Type	Principal
Well Known	No, although this value is hardcoded in the configuration scripts. To change this value, you must edit the configuration scripts.
Description	A DTS Server on node *hostname* runs as this principal. This principal must be a member of **/.:/subsys/dce/dts-servers**.
Initial Configuration ACL	
Object ACL	**Default cell = /.../*cellname*** **unauthenticated:r--------** **user_obj:r---f--ug** **user:*creator*:rcDnfma-g** **group:acct-admin:rcDnfma-g** **group:subsys/dce/dts-admin:rcDnfma-g** **other_obj:r-------g**
Created By	DCE configuration

Name	**/.:/sec/principal/hosts/*hostname*/cds-server**
Security Type	Principal
Well Known	No
Description	A CDS Server on node *hostname* runs as this principal. This principal must be a member of **/.:/subsys/dce/cds-server**.
Initial Configuration ACL	
Object ACL	**Default cell = /.../*cellname*** **unauthenticated:r--------** **user_obj:r---f--ug** **user:*creator*:rcDnfmaug** **group:acct-admin:rcDnfma-g** **group:subsys/dce/cds-admin:rcDnfma-g** **other_obj:r-------g** **any_other:r--------**
UNIX UID	Generated
Created By	DCE configuration

Name	**/.:/sec/principal/hosts/***hostname***/dfs-server**
Security Type	Principal
Well Known	No
Description	This is the principal name of the DFS Servers.
Initial Configuration ACL	
Object ACL	**Default cell = /.../***cellname* **unauthenticated:r-------g** **user_obj:r---f--ug** **user:creator:rcDnfmaug** **group:acct_admin:rcDnfma-g** **other_obj:r-------g** **any_other:r--------**
UNIX UID	Generated
Created By	DCE configuration

Name	**/.:/sec/principal/hosts/***hostname***/gda**
Security Type	Principal
Well Known	No
Description	The GDA on node *hostname* runs as this principal. This principal must be a member of **/.:/subsys/dce/cds-servers**.
Initial Configuration ACL	
Object ACL	**Default cell = /.../***cellname* **unauthenticated:r-------g** **user_obj:r---f--ug** **user:***creator***:rcDnfmaug** **group:acct-admin:rcDnfmaug** **group:subsys/dce/cds-admin:rcDnfmaug** **other_obj:r-------g** **any_other:r--------**
UNIX UID	Generated
Created By	DCE configuration

Name	**/.:/sec/principal/hosts/*hostname*/self**
Security Type	Principal
Well Known	Yes
Description	This entry is the principal for host *hostname*. The **sec_clientd** daemon uses this principal. This can also be the principal that child processes of the **init** command use.
Initial Configuration ACL	
Object ACL	**Default cell = /...**/*cellname* **unauthenticated:r--------** **user_obj:r---f--ug** **user:***creator***:rcDnfma-g** **group:acct-admin:rcDnfma-g** **other_obj:r-------g** **any_other:r--------**
UNIX UID	Generated
Created By	Security configuration

Appendix B

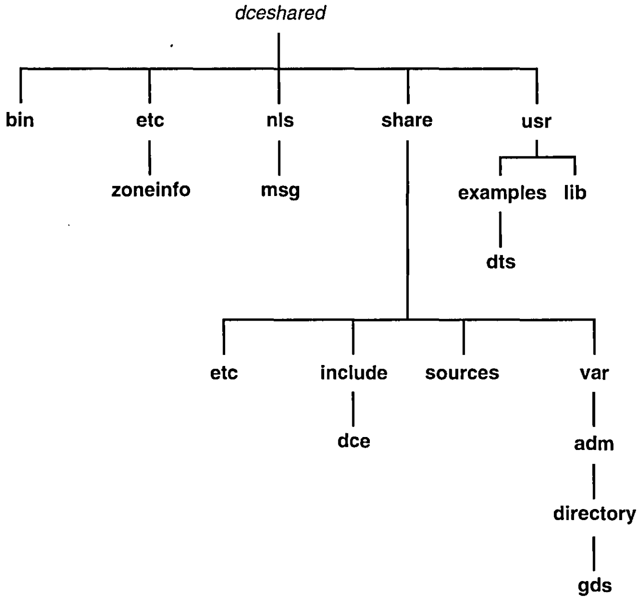
The Location of Installed DCE Files

This appendix shows the organization of the *dceshared*, *dcelocal*, and the UNIX subdirectories that are used by DCE.

B.1 The dceshared Subdirectories

Figure B-1 shows the *dceshared* subtree.

Figure B–1. The *dceshared* Subtree



The following directories are created in the *dceshared* subtree during installation:

- *dceshared/***bin**

  This directory contains utilities for applications programmers and DCE users, DCE administration utilities, and server processes (daemons).

- *dceshared/***etc**

  This directory contains templates of configuration files that are in architecture-dependent format.

- *dceshared*/**etc/zoneinfo**

  This directory contains templates of configuration tables.

- *dceshared*/**nls/msg/${LANG}**

  This directory contains delivered message catalogs (***.cat**) files for each supported language.

- *dceshared*/**share**

  All of the previously described subdirectories can contain architecture-dependent files, which are addressable by using **@sys**. However, the files that are listed after *dceshared*/**share** are completely architecture independent.

- *dceshared*/**share/etc**

  This directory contains templates of common (shared) configuration files.

- *dceshared*/**share/include**

  This directory contains application header files and DCE internal header files. The **/usr/include/dce** directory is linked to this entire directory, but in future DCE releases it could be separated and linked only to those files that are necessary for writing DCE-based applications.

- *dceshared*/**share/sources**

  This directory contains DCE sources and build tools as organized in the Open Development Environment (ODE) build tree, which is available to DCE source licensees only.

- *dceshared*/**usr/examples**

  This directory contains example executable files.

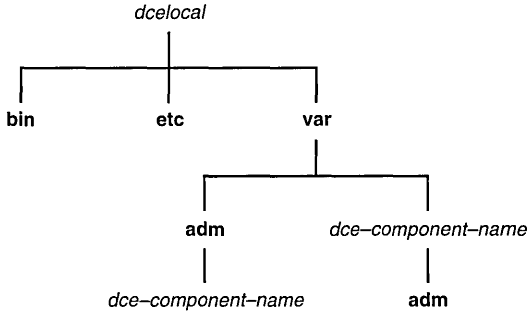
- *dceshared*/**usr/lib**

  This directory contains application libraries (**libdce.a**) and DCE internal libraries.

B.2 The dcelocal Subdirectories

Figure B-2 shows the *dcelocal* subtree.

Figure B–2. The *dcelocal* Subtree



The following directories are created in the *dcelocal* subtree during installation:

- *dcelocal*/**bin**

  This directory contains DCE administration utilities and server processes (daemons), which are necessary for local client system initialization and for server machines.

- *dcelocal*/**etc**

  This directory contains local (machine-private) configuration files, which are maintained by client machines. This directory has write permission for the local system administrator only.

- *dcelocal*/**var**/**adm**/*dce-component-name*

  This directory contains log files (including core images) and cache files, which are maintained by client machines. For convenience, symbolic links from **/var/adm/dce/client**/*dce-component-name* are created. This directory has write permission for the local system administrator only.

- *dcelocal*/**var**/*dce-component-name*

  This directory contains all data files (configuration files, databases, and so forth), which are maintained by each of the DCE servers. To provide high availability and, in case of the Security Service, appropriate protection, data files that are associated with a service are usually physically located at the server site. Therefore, they are stored in separate trees under *dcelocal*/**var**.

Files in *dcelocal*/**var**/*dce-component-name* are only those that are accessed by dedicated servers. This directory has write permission for the service administrator only.

Configuration and log files that are relative to client machines are not stored here. These files are in *dcelocal*/**etc** and *dcelocal*/**var**/**adm**.

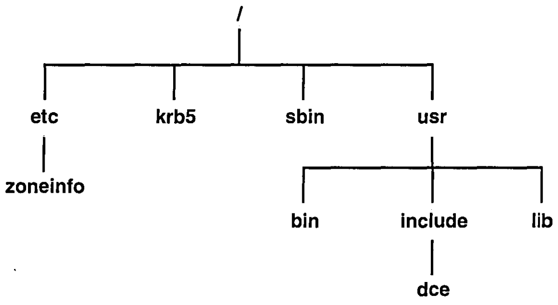
- *dcelocal*/**var**/*dce-component-name*/**adm**

  This directory contains server log files and cache files, which are maintained by server machines. This subdirectory needs to be maintained by each service to store the log and cache files. Because users sometimes expect log files in conventional locations, /**var**/**adm**/**dce**/*dce-component-name* is created as a symbolic link to these directories. This directory has write permission for the service administrator only.

B.3 Conventional UNIX Directories

Figure B-3 shows the directories that DCE uses in the standard UNIX tree.

Figure B–3. Standard UNIX Directories Tree



DCE uses the following standard UNIX directories.

- **/etc/zoneinfo**

  This directory contains copies of the templates, which are modified, if necessary, from the *dceshared*/**etc/zoneinfo** directory.

  Note: Preexisting files can be modified on the local system. Be careful not to overwrite them during the installation procedure.

- **/krb5**

  This directory contains Kerberos configuration files for the conventional **krb5** environment. Symbolic links exist to appropriate files in *dcelocal*/**etc**. This directory has write permission for the local system administrator only.

- **/sbin**

  This directory contains the small set of executables, which are derived and copied from the *dceshared*/**bin** directory, that are required in the root partition. Although this is an exact subset of the executables that are found in the *dcelocal* directory, you need to keep a copy in *dcelocal*/**bin** because **/sbin** is usually not set as part of **$PATH** on running systems. This is only applicable for DCE Diskless clients.

- **/usr/bin**

  This directory contains utilities for applications programmers and DCE users. Most of these are symbolic links that point to *dceshared*/**bin.** Some utilities, such as **login** and **su**, can actually be local copies that are needed for performance and high availability. On server machines, copies of the respective executables are sometimes necessary for the initialization of the system.

- **/usr/include/dce**

  This directory contains DCE header files. This directory is a symbolic link to *dceshared*/**share/include/dce**.

- **/usr/lib**

  This directory contains **libdce.a**, which is a symbolic link to *dceshared*/**lib/libdce.a.**

Index

OPEN SOFTWARE FOUNDATION™

INFORMATION REQUEST FORM

Please send to me the following:

( ) OSF™ Membership Information

( ) OSF™DCE License Materials

( ) OSF™DCE Training Information

Contact Name ______________________________

Company Name ______________________________

Street Address ______________________________

Mail Stop ______________________________

City ____________ State _____ Zip ________

Phone ____________ FAX ____________

Electronic Mail ______________________________

MAIL TO:

Open Software Foundation
11 Cambridge Center
Cambridge, MA 02142

Attn: OSF™DCE

For more information about OSF™DCE call OSF Direct Channels at 617 621 7300.

OSF™DCE

OSF™DCE Administration Guide — Introduction

TITLES IN THE OSF™DCE SERIES:

Introduction to OSF™DCE

OSF™DCE User's Guide and Reference

OSF™DCE Administration Guide
— Introduction
— Core Components
— Extended Services

OSF™DCE Administration Reference

OSF™DCE Application Development Guide

OSF™DCE Application Development Reference

Application Environment Specification (AES)
Distributed Computing

Printed in the U.S.A.

Open Software Foundation
11 Cambridge Center
Cambridge, MA 02142

Prentice Hall, Inc.

ISBN 0-13-176546-9

90000

9 780131 765467

DISTRIBUTED SYSTEMS

OSF™DCE

OSF™DCE Administration Guide — Introduction

TITLES IN THE OSF™DCE SERIES:

Introduction to OSF™DCE

OSF™DCE User's Guide and Reference

OSF™DCE Administration Guide
— Introduction
— Core Components
— Extended Services

OSF™DCE Administration Reference

OSF™DCE Application Development Guide

OSF™DCE Application Development Reference

Application Environment Specification (AES)
Distributed Computing

Printed in the U.S.A.

Open Software Foundation
11 Cambridge Center
Cambridge, MA 02142

Prentice Hall, Inc.

ISBN 0-13-176546-9

90000

9 780131 765467

DISTRIBUTED SYSTEMS