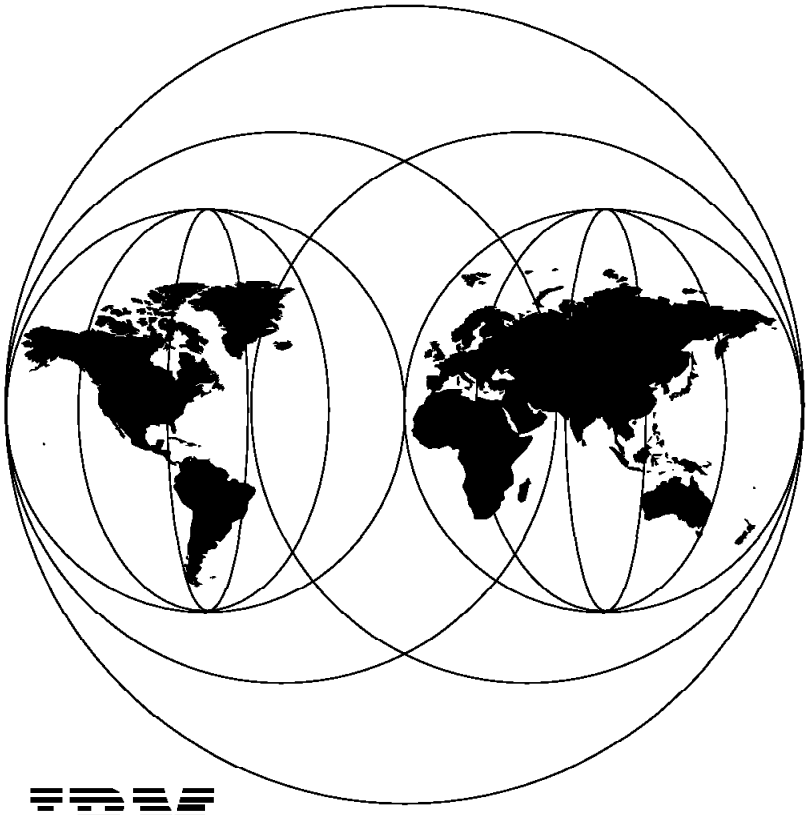
SG24-4746-00

DCE Cell Design Considerations

June 1996



IBM

International Technical Support Organization
Poughkeepsie Center

International Technical Support Organization

SG24-4746-00

DCE Cell Design Considerations

June 1996

Take Note!

Before using this information and the product it supports, be sure to read the general information in Appendix C, “Special Notices” on page 97.

First Edition (June 1996)

This edition applies to:

- Version 2 Release 1 of DCE Product Family, 5765-533 IBM DCE Security Services, 5765-534 IBM DCE Cell Directory Services and 5765-537 IBM DCE Enhanced Distributed File System for use with AIX Version 4 Release 1
- Features OpenEdition DCE Base Services (OSF DCE level 1.1) and OpenEdition DCE Distributed File Service (OSF DCE level 1.0.3a) for use in 5645-001 OS/390 Version 1 Release 1
- Beta Version of DCE Cell Directory and Security Services in 5622-851 OS/2 Warp Server, Version 4

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. HYJ Mail Station P099
522 South Road
Poughkeepsie, New York 12601-5400

When you send information to IBM, you grant IBM a non-exclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

Contents

Figures

Tables

Preface

DCE is a set of software components from the Open Software Foundation (OSF) that provides a platform for the construction and use of distributed applications. This redbook provides details on how to design connections and how to perform administrative functions that result in a working administative DCE domain, called a cell. The book focuses on considerations of DCE cell design. It provides information about determining whether single-cell or multiple-cell designs are appropriate, placing administrative and application server functions, choosing hardware and operating system resources, and integrating a DCE approach with a strategy to use Internet facilities. The particular operating systems that are shown in examples and descriptions of administration are MVS, AIX/6000 and OS/2, although reference to others is included.

This redbook was written for network and application designers, consultants, service providers, and other technical professionals. Some knowledge of DCE and distributed applications is assumed.

How This Redbook Is Organized

This redbook contains 112 pages. It is organized as follows:

- Chapter 1, “Introduction”

  This provides an introduction to the terminology and outline for our project. Included is a general overview of DCE, and descriptions of the architecture of the DCE components.

- Chapter 2, “Cell Design - General Considerations”

  This chapter outlines the specific considerations of DCE cell design. There is discussion of how to determine whether to use a single cell or to define multiple cells. Elements involved in the choice of the design range from business and organization considerations to hardware and network configurations.

- Chapter 3, “Sample Scenarios - Theoretical and Real”

  This chapter describes and compares theoretical cell designs. Consequences on DCE servers and cell administration are discussed. In addition, a DCE implementation that is often used in real production is described.

- Chapter 4, “DCE Server Considerations”

  This chapter describes DCE cell configuration on different IBM platforms. Examples for first installation steps are included, as well as client and server definitions. Specific definitions are outlined that relate to the multiple cell environment.

- Chapter 5, “Application Implications”

  This chapter provides more detail on the DCE application servers. Application support server on OS/390, the application development process in general, and tools that can be used to help in that process are described. IBM applications that use DCE are reviewed.

- Chapter 6, “DCE on the Internet”

This chapter describes how DCE can be used in applications with Internet access. Specifically, two areas are considered:

1. Allowing customers from the Internet to access DCE applications.
2. Allowing DCE cell members to connect to a cell by using the Internet.

The Team That Wrote This Redbook

This redbook was produced by a team of specialists from around the world working at the International Technical Support Organization Poughkeepsie Center.

Jan Baisden is a Marketing Support Representative at the International Technical Support Organization, Poughkeepsie Center. He writes extensively on areas of AIX, Open Systems, and DCE. Before joining the ITSO two years ago, he worked in the IOSC, Gaithersburg as a Marketing Support Representative.

Holger Bruhn is a Systems Engineer for AIX Support in Germany. His areas of expertise include AIX-DCE, AIX, and SP2. He has worked at IBM for 26 years.

Jean-Claude Jesionka is a Architect in Information Systems in France. He has worked at IBM for 15 years. His areas of expertise include DCE and MVS.

Thanks to the following people for their invaluable contributions to this project:

Michel Plouin
International Technical Support Organization, Poughkeepsie Center

Peter Wassel
IBM Glendale Labs, Endicott, New York

Mary Hu
IBM US, Gaithersburg, Maryland

Comments Welcome

We want our redbooks to be as helpful as possible. Should you have any comments about this or other redbooks, please send us a note at the following address:

redbook@vnet.ibm.com

Your comments are important to us!

Chapter 1. Introduction

This chapter provides background and definitions so that understanding of the later chapters is easier.

1.1 DCE Overview

This section provides some general information about DCE, the history, the advantages and the different system platforms supporting DCE.

1.1.1 What Is DCE?

DCE is the Distributed Computing Environment, from the Open Software Foundation (OSF); it is an architecture, a set of standard services, and application program interfaces (APIs), used to support the development and usage of distributed applications.

DCE consists of multiple components that have been integrated to work closely together. They are the Remote Procedure Call (RPC), the Cell Directory Service (CDS), Global Directory Services (GDS), the Security Service, DCE Threads, Distributed Time Service (DTS), and Distributed File Service (DFS). The Threads, RPC, CDS, Security, and DTS components are commonly referred to as the "secure core" and are the required components of any DCE installation. DFS is an optional component. DCE also includes administration tools to manage these components.

DCE is called "middleware" or "enabling technology." It is not intended to exist alone, but instead should be bundled into a vendor′s operating system offering or integrated into that offering by a third-party vendor. DCE′s security and distributed file system, for example, can completely replace their current, non-network analogs. DCE is not an application in itself, but is used to build custom distributed applications or to support purchased applications.

1.1.2 How DCE Was Created

The Distributed Computing Environment can trace its beginning to the Open Software Foundation (OSF). DCE was defined by OSF to include mechanisms for communication between parts of distributed applications, a way for these parts to find each other, security services for this applications, and more. OSF has now evolved into an organization called The Open Group. The Open Group will continue to develop the architecture and components of DCE as technoligies expand and experience is attained.

Various major computer vendors have contributed their expertise and proven technologies for inclusion in DCE. DCE 1.0 was released by OSF in January 1992. DCE is now available on all major operating system platforms.

1.1.3 What Are the Advantages of DCE?

First, DCE provides services that can be found in other computer networking environments, but packages them so as to make them much easier to use. For example, the DCE Remote Procedure Call (RPC) facility provides a way of communicating between software modules running on different systems that is much simpler to code than older methods, such as using socket calls.

Second, DCE provides new capabilities that go beyond what was available previously. For example, the DCE Security Service provides a reliable way of determining if a user of a distributed system should be allowed to perform a certain action. This is very useful for most distributed applications, yet the design and implementation effort entailed in providing such a capability would be prohibitive for an individual developer.

Third, DCE integrates components in a manner that makes them more valuable together than separately. For example, the DCE RPC uses threads in such a way that a developer can implement a multi-threaded server without ever explicitly creating or destroying a thread.

Fourth, DCE supports both portability and interoperability by providing the developer with capabilities that hide differences among the various hardware, software, and networking elements that an application will deal with in a large network. For example, the RPC automatically converts data from the format used by one computer to that used by another. Portability is a measure of the ease with which a piece of software that executes on one type of computer can be made to execute on a different type of computer. Interoperability is a measure of the ability of computers of different types to participate in the same distributed system.

Finally, DCE supports (as an optional component), the Distributed File Service (DFS). This is a means of making files available to a network that can be used as if they are locally present on workstations in that network. Details of DFS will be provided in later sections.

1.1.4 What Platforms Support DCE?

DCE is supported on most UNIX platforms and many non-UNIX platforms. Most vendors support at least the “secure core,” a term which refers to all of the DCE services except the Distributed File Service and X.500 interface to the Global Directory Service.

Some products are client-only, which means that the following servers for the DCE services are not provided: Directory Service, Security Service, Time Service. Client machines can use these services, but they cannot run the server programs. Another machine in the cell must run the server programs. Application programs can be built and application servers can be run on these “client-only”systems.

The following table (Table 1) shows the DCE implementations available on IBM operating systems:

Table 1. OSF/DCE 1.1 Implementations on IBM Platforms

System	**Comment**
AIX	DCE/DFS client, CDS, SEC, DTS, DFS, GDS server
OS/2 Warp Server	DCE/DFS client, CDS, SEC, DTS server
VM/ESA	DCE client
MVS/ESA 5.2.2	DCE client, DTS, DFS (OSF 1.0.3a) server
OS/390	DCE client, SEC, DTS, DFS (OSF 1.0.3a) server
IBM AS/400	DCE client (1H96)

In Figure 1 on page 3 you can see the different system types that may reside in a DCE cell and possible roles they may perform.

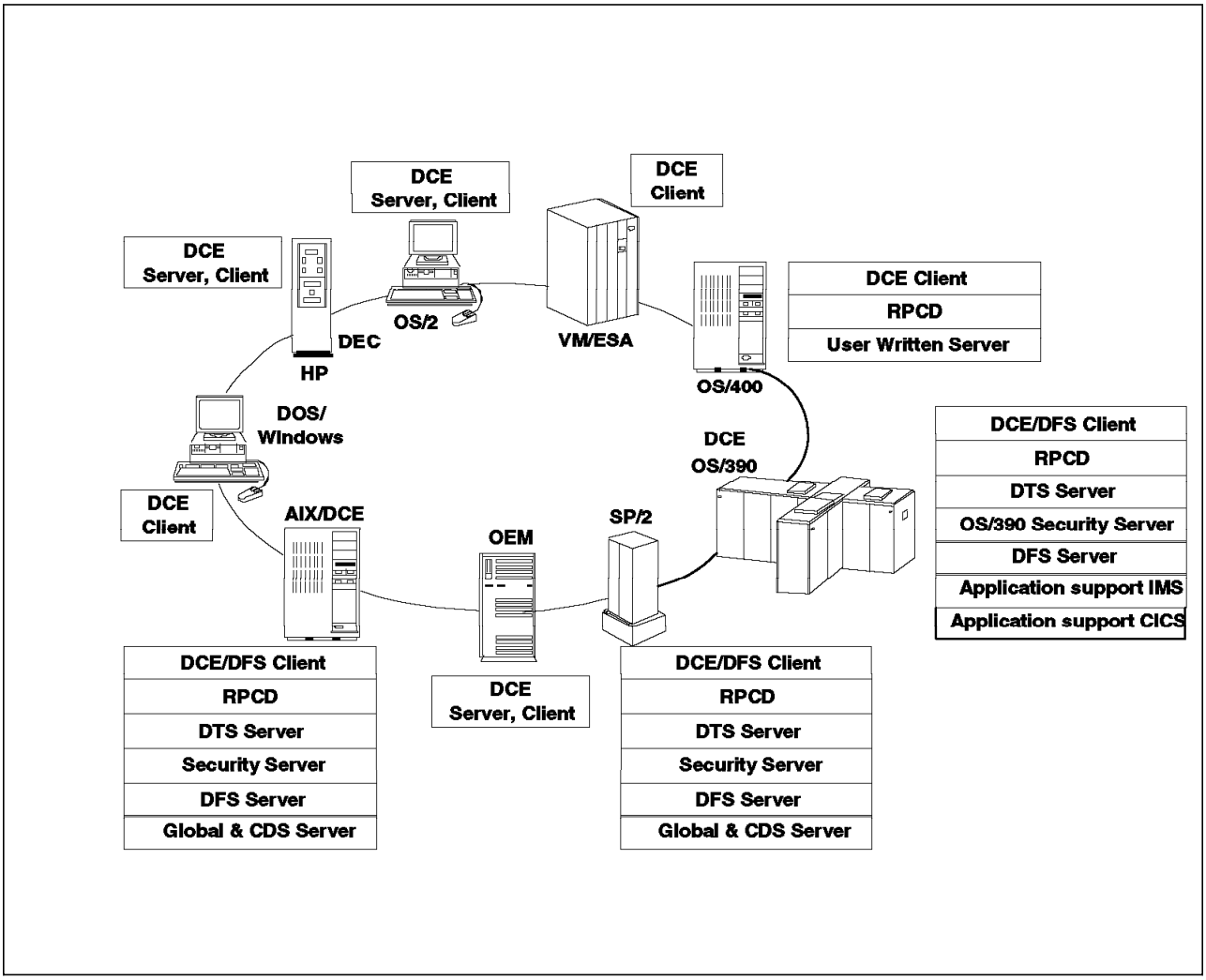


Figure 1. DCE - Platforms Supported. DCE DFS client on OS/390 is an announced support, to be available in 1996.

1.2 DCE Architecture

This section describes the interconnected operation of the DCE components. Included here is a description of the operational framework of DCE.

1.2.1 DCE Cells

A cell is DCE′s basic unit of organization. It is an administrative domain that allows that allows those users, machines, and resources that share a common purpose to be managed through functions distributed within the network in which they exist. Members of an organization who are working on the same project are likely to belong to the same cell. In a large organization with several cells, the sales team could belong to one cell and the finance team could belong to a second. A small organization might only have one cell for both sales and finance because they share the same information and the same level of security.

Members of a cell are usually located in a common geographic area, but they could be located in different buildings, different cities, or even different countries, if good communications facilities exist.

1.2.2 DCE Components and Services

The following topic gives a short description of the DCE components. For further Information and more details refer to the section Appendix D, “Related Publications” on page 99 where you can find a listing of the available DCE documentation.

Figure 2 gives an general overview about the different components of DCE, their relations and how they fit together.

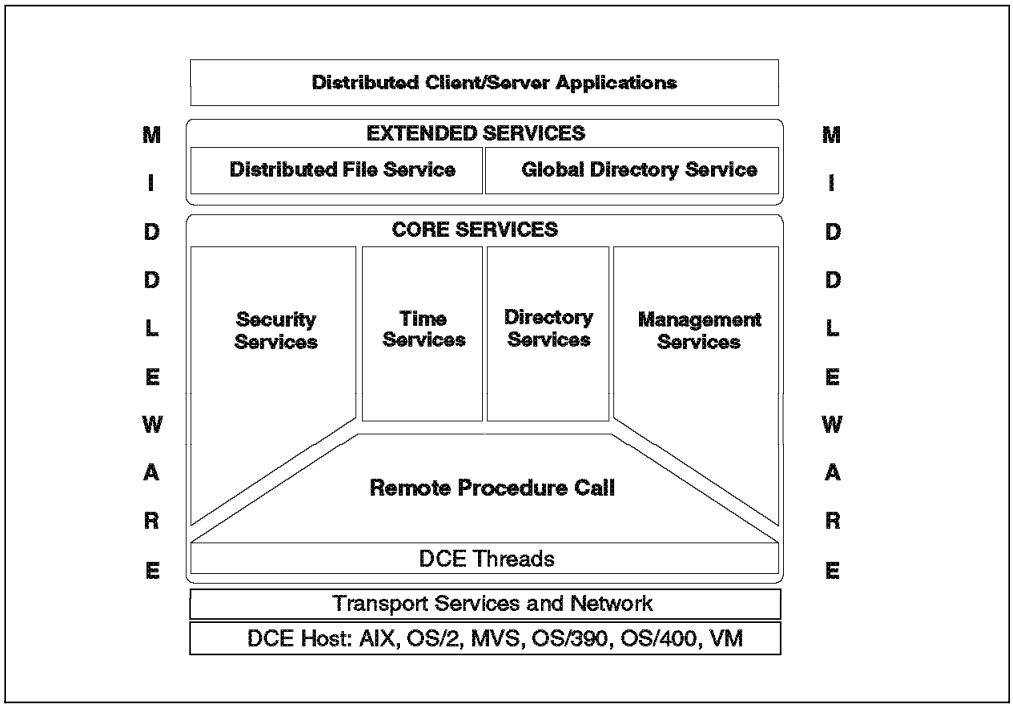


Figure 2. DCE Architecture

1.2.2.1 DCE Remote Procedure Call

The DCE Remote Procedure Call (RPC) is a service for calling a procedure on a remote machine as if it were available on the same machine where the call originates. Based on the client/server model, it allows application programmers to extend the local procedure call to a distributed environment. The application programmer does not have to be concerned with the details of network communications between client and server nodes.

Programmers using RPC do not need to rewrite applications to port them to different architectures, operating systems or communication protocols. RPC hides communication details and removes system and hardware dependencies.

An end user does not see any of the client/server interaction and does not know or need to know if procedures are local or remote.

1.2.2.2 DCE Threads

Many computer programs are designed to execute sequentially, with only one point in the program currently in execution. However, some computer programs lend themselves to being structured with multiple flows of control, and perform better when they contain multiple threads of control.

In distributed computing, RPC enables the use of multiple threads of control. When a client issues an RPC, it blocks (or waits) until a response is returned from the server. If a client uses multiple threads of control, work can continue in another thread while the thread awaiting an RPC response is blocked. Because servers can also issue RPCs, a similar scenario applies. A separate thread can handle each client request. While one server thread is waiting for an input or output operation to finish, another server thread can continue working.

1.2.2.3 DCE Directory Service

A distributed system may contain many users, machines, and other resources, along with large amounts of data; all these could be geographically dispersed. The goal of a directory is to provide up-to-date addressing information for network resources. Users can identify, by name, resources such as servers, files, disks or print queues, and gain access to them without needing to know where they are located. The sharing of information is based on unique names, not on location.

The DCE Directory Service is a distributed and replicated service. It is distributed because the information that forms the database is stored in different places. Information on one group of users and resources can be stored on one directory server, while information about a second group of users and resources is stored on a different directory server. The directory service can replicate information, storing it in more than one location, to make it more readily available. Performance is enhanced because directory information can be replicated close to its users.

The directory service consists of two types of directory services: one that manages resources within a cell, and a global name service that provides user access to servers in outside cells.

The Cell Directory Service (CDS) stores and manages the names and attributes of resources in a cell. There are two global name services that can be used to locate resources outside the cell: Global Directory Service (GDS) and Domain Name System (DNS). GDS is based on the CCITT X.500/ISO 9594 international standard, and so is positioned to participate in the anticipated worldwide X.500 directory service. DNS, used by the Internet to track the vast number of machines that comprise this “network of networks” provides an alternative means for independent cells to locate and interact with each other.

1.2.2.4 DCE Distributed Time Service

The DCE Distributed Time Service (DTS) provides time synchronization for the computers participating in a distributed computing environment. In a single system, one clock provides the time to all applications. In a distributed system each node has its own clock. Even if all the clocks in a distributed system could be set to one consistent time at some point, they would drift away from that time at different rates. As a result, different nodes would have different times. This is a problem for distributed applications where the ordering of events is important. DTS enables distributed applications to determine event sequencing, duration

and scheduling. DTS synchronizes a DCE host's time with Universal Coordinated Time (UTC), an international time standard.

1.2.2.5 DCE Security Services

The DCE Security Service provides trustworthy identification and certification of principals (users, clients, servers and systems). It offers integrity and privacy of communications and enables controlled access to resourses. It controls the interaction between clients and servers.

Today, most systems provide one way authentication, where the client proves its identity to the server. Server identity is rarely verified. In a distributed environment, this trust of servers may be lessened, leading to a requirement for two-way authentication.

In two-way authentication, each server must be able to verify the identity of each client. For its part, the client must be confident that it is communicating with a secure server. Clients and servers use trusted keys to request and provide services. Each server must maintain trusted key information for each client that it can serve, and every client must know a trusted key for each server it might use. Two-way authentication is difficult to administer. Every time a server's information changes, all the clients must be updated.

DCE simplifies administration and adds security by implementing a trusted-third-party approach based on the Kerberos technology. The security server, acting as a trusted third party, maintains the trusted key information. Clients and servers no longer need to store this information. The security server identifies and certifies principals (authentication) and provides information on the privileges associated with each principal. Privileges enable servers to perform selected operations (authorization) for authenticated principals.

1.2.2.6 Distributed File Service

The DCE Distributed Files Service (DFS) allows users to access and share files stored on a file server anywhere on the network, without having to know the physical location of the file. Files are part of a single namespace. Therefore, no matter where in the network a user is, the file can be found using its unique name.

DCE DFS includes a physical file system, called the DCE Local File System (LFS), which supports special features that are useful in a distributed environment. DCE LFS gives you the ability to perform the following tasks:

- Replicate data.
- Log file system data, enabling quick recovery after a crash.
- Simplify administration by dividing the file system into easily managed units called filesets.
- Associate Access Control Lists (ACLs) with files and directories; ACLs enable control of user access privileges.

With DCE Version 2.1 for AIX, IBM has enhanced DFS to allow users to export an AIX CD-ROM file system from a DFS File Server. The exported CD-ROM file system can be mounted into the DFS file space an accessed from DFS client machines.

The local file system application programming interfaces (APIs) are used for DFS, no separate interfaces are used. DFS administration is performed by using commands.

1.3 The IBM Open Blueprint and DCE

A major goal of the Open Blueprint is to enable a single-system view of the network, masking the complexities of the physical network environment from users. It serves four major roles:

- Helps customers develop their own architectures and organize products and applications in an open environment
- Describes IBM's direction for products and solutions in the open distributed environment
- Guides developers as they supply products that include the appropriate functions and products that can be integrated and interoperate with other installed products
- Provides a context for incorporating new technologies into a distributed environment

IBM has worked closely with international standards organizations (ANSI, IEEE, ISO), industry consortia (X/OPEN, Object Management Group (OMG), OSF), customers, and industry leaders to determine the most widely accepted standards. These standards allow IBM products to interact with each other and with other vendor products.

The Open Blueprint describes the layers and functions. The Open Blueprint structure allows a network of operating systems to function as a unit. Different layers build the functions.

Note

Distributed Computing Environment DCE from OSF plays a central role in the Open Blueprint.

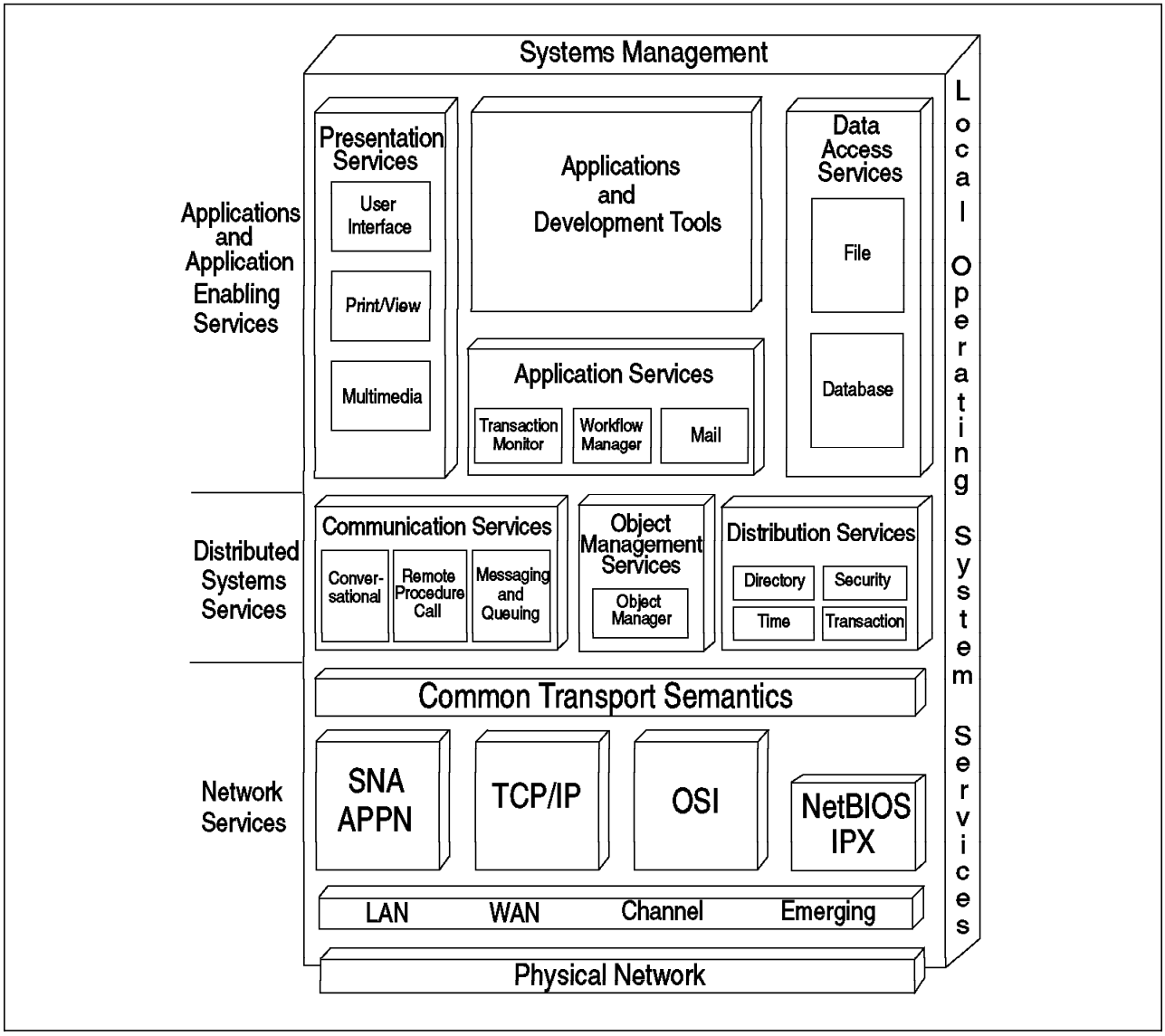


Figure 3. Open Blueprint

We review the three areas of the Open Blueprint that are directly related to DCE implementation:

- Communication services
- Distribution services
- Data access services

Communication services and Distribution services are part of Distributed Systems Services (DSS). DSS enables applications to locate other applications in the network and conduct a secure and reliable communication independent of the underlying network.

1.3.1 Communication Services

The Open Blueprint supports three models that describe how parts of the distributed applications or services cooperate or communicate with each other.

Some types of applications are more suited to one model than another. An application can use more than one model. Each model uses the directory and security services to find the part requested and to verify that the access is authorized. They are:

- Conversational Communication
- Remote Procedure Call
- Messaging and Queuing

RPC is the base communication component of DCE.

1.3.2 Distribution Services

The Distribution Services provide a single-system view of the network and consist mainly of the DCE core services. The functions are:

- The Directory and Naming Services
- The Security Service
- The Time Service

1.3.3 Data Access Services

The *Data Access Services* addresses access to files and databases. The file service APIs define how record-oriented or byte-stream files are accessed or how entire files are transferred. The standard for the byte-stream API is the DCE Distributed File System (DFS). The Open Blueprint also defines that any file is accessible through any protocol, meaning, for instance, DFS will be accessible through gateways from NFS or Netware and so on.

The database service supports access to relational, object-oriented and hierarchical databases. For instance, access to relational databases is SQL, and the connectivity to remote relational databases is defined in the Distributed Relational Database Architecture (DRDA). IBM has announced DRDA support over TCP/IP, allowing DB2 to use DCE security functions.

1.3.4 Network Services

Network services is not an implementation of DCE but simplifies it by making protocols independent of the transport layer.

The *Common Transport Semantics* (CTS) insulates the applications and the higher-level services of the Open Blueprint from the underlying network transport by providing a common view of transport protocols, thus making the applications transport-independent. Using CTS also enables integration of networks with different protocols through transport gateways. In the Open Blueprint, CTS is provided through the Multiprotocol Transport Networking (MPTN) architecture.

IBM has developed the AnyNet products (based on MPTN architecture) that shield requesters from network differences. With AnyNet, you can tunnel SNA protocols through TCP/IP or vice versa.

In the case of MVS, the usual situation is to find a S/390 machine connected to an SNA network. Once TCP/IP is installed on MVS to support DCE applications,

the Anynet feature of VTAM allows the transport of the TCP/IP protocol over the SNA network. The legacy applications can be accessed through a DCE infrastructure without major network changes.

Chapter 2. Cell Design - General Considerations

Today more and more applications are developed over a distributed computing infrastructure, that is, over a two-tier or three-tier architecture. Some are new applications, but also there are old applications involved in a re-engineering process.

DCE technology brings solutions to many of the technical issues that arise in a distributed computing configuration, but planning and design for the distributed computing configuration is essential to achieve a successful application implementation.

In this chapter, we look at some of the elements that a designer must consider when planning for a new DCE configuration. The challenge for the designer is to take all these elements into account and to apply DCE technology to answer most, and if possible all, user needs and application requirements.

These are some of the elements (general and technical) that the cell designer must take into account:

- Customer and business needs
- Applications requirements
- Organization and enterprise structure
- Network and systems layout
- Administration policies and tools
- Security Requirements and Administration
- Cost

Additional subjects could be added to this list on a personal experiment basis. We hope that the topics we develop in this book cover most of the subjects the designer should consider.

We discuss in this chapter how each of these elements may influence the design of a DCE cell and what leads you to select a single cell design or a multiple cell design.

2.1 Multiple Cell Definition

A definition for cell as used in DCE is given in 1.2.1, “DCE Cells” on page 3. A multiple cell environment is a configuration where a user from his own cell will have some access to resources in other cells. Such a design requires cross-cell administration to establish adequate communications between the cells. The main tasks are:

- Establish network communication between the cells.
- Establish links between the directories of each cell through *Global Directory Services (GDS)* or *Domain Name Services (DNS)* and the *Global Directory Agent (GDA)*.
- Create cross-cell authentication accounts in the registry database of each cell.
- Create authorizations for DFS usage across the cells.

- Manage the foreign attributes of namespace and other DCE object ACLs.

All these tasks are described in more detail in 4.6, “ Multiple Cell Definitions” on page 67.

Each administrator keeps control over his own cell, but an overall administrative role is needed to insure consistency between the cells for communications between them.

2.2 Customer and Business Needs

We discuss in this chapter the early steps for a distributed environment project.

Customer and business needs must be established as clearly as possible because they are the starting point of any development project. Several consulting methods and tools are available from IBM to assist the cell planner. They will help him to check if the proposed design is consistent with the goals of the enterprise.

The planner should use a method such as *Joint Application Design (JAD)* which delivers the main guidelines of the new application to the customer. Consistency of these guidelines with the overall strategy of the enterprise is validated.

A tool, the *Client/Server Advisor System*, includes several components that allow a designer:

To assess the business and technical environment

To identify potential areas of improvement

To create a conceptual design

These are only examples among the large set of service offerings delivered by IBM in the field of distributed computing.

IBM uses and provides under license a comprehensive methodology for the planning and the management of projects. The methodology is named *IBM Application Development and Systems Integration Methodology (IBM AD/SI-M)*. This methodology will allow the consideration of the application and system design framework, toward which the cell designer will focus his DCE knowledge.

2.3 Application Requirements

Good performance and reliability are the goals of any application, and we will see how DCE technology achieves these goals.

2.3.1 General Considerations

The cell design guidelines must consider client expectations and needs in the areas of availability and performance. This will have a strong impact on the localization of each of the services used by the client:

- Application services
  - Transaction services
  - Database services
  - File services
  - Print services

- Technical services
  - Security services
  - Directory services

DCE technology mainly uses replication techniques to improve the performance and availability of the servers in a cell. The definition of the number of replicas and replica location will be the main administrative task when building a reliable and well-tuned cell. The capability to create and maintain these replicas in a multiple cell design will determine if the configuration is robust and well-managed.

Another aspect of DCE technology is server selection. Most of the time, the client does not choose which directory server or replica it will search first, nor which security server it will contact first. DCE internal mechanisms are based on:

- Localization priorities - the search for a DFS FLDB will be done on the local server first, then the subnetwork, then the network, and finally other networks.
- Voting processes - at least three servers of the same DTS or DFS component of DCE are required in order to to make a decision on which of them will be the master. DTS clients use this process, however localization priority is also used in that an attempt to locate servers on the local LAN is made first. If enough servers are found there, more remote locations are not consulted. Obviously, if servers are found on the local LAN, performance is better.
- Random search - CDS clearinghouse and security servers are searched randomly.

Considering these mechanisms, which cannot be totally controlled from an administrative point of view, the cell designer and the cell administrator may have to review the server locations within a cell to improve application performance.

Consideration should also be given to the currency of the data available for an application. If the data changes frequently then all the replicas have to be updated. So replication, which has been chosen for better performance and availability, may generate high network traffic overhead to keep data up-to-date. The logical conclusion is: if data changes frequently a single central location solution will provide better performance than a solution involving replicas.

2.3.2 Servers Location

The application's requirements in terms of performance will always lead the cell designer to locate application servers *in the same cell* as the users of this application. This will avoid going through cross-cell name resolution and caching the server location (not caching would mean performing resolutions on each request). In a single cell design, application requirements will lead the designer to focus on locating replicas to achieve good workload balance and good reliability of the cell.

In a multiple cell environment, if an application is to be used by clients from several different cells, the implementation can be done in two different ways:

1. The application server is in one cell (the one that has the highest number of users of this application) and is accessed from another cell using GDS or DNS name resolution and cross cell authentication. This is the case when the application server is "OpenEdition DCE Application Support" for IMS or

CICS on one S/390 system. Most high usage application servers will be done this way.

2. The application server is duplicated in every cell. This adds new administrative tasks to keep all the application servers at the same level of currency.

Note: "OpenEdition DCE Application Support for MVS/ESA" provides a specialized DCE server on MVS for transaction processing. The server is shipped with an IMS feature, a CICS feature, or an IMS/CICS feature. A DCE client request is handled and converted to a form understandable by an IMS or a CICS region. From the perspective of DCE clients, IMS or CICS regions appear as regular RPC-based servers. This product is described in more detail in *OS/390 OpenEdition DCE Application Support for MVS/ESA Configuration and Administration Guide Release 2*, SC09-1659-00. An ITSO redbook also gives more implementation details. That book is *MVS/ESA OpenEdition DCE: Application Support Servers CICS and IMS*, GG24-4482. See also 5.2.1.2, "DCE Application Support for MVS/ESA" on page 72.

2.4 Existing Organization

The ideal situation is to be able to design a cell from scratch and map it exactly to the needs of the new business that is created. This freedom of decision may also happen in the case of re-engineering a business process if the choice is to implement new computing facilities with little or no migration from existing ones. In most cases however, existing organizations and structures will have a strong impact on the design of the cell.

2.4.1 Types of Organization

The business of the company, its organization, and its processes, determine the data flow that is supported by the Information Technology (IT) system. An analysis of the data flow provides a good basis for DCE cell design.

A retail company or a bank is usually organized in a headquarters structure (in one or more locations) and a branch office structure. The characteristics of such an organization is that all the locations have the same profile and only the applications vary. In this case the data flow has two components:

- A local component in the branch office or the headquarters for local applications (sales, retail banking operations, printing, and so forth)
- A cross company component from the branch office and to the headquarters (results consolidation, administrative information)

These two data flows have approximately the same importance.

For this type of profile *(organization type 1)*, we should rather recommend a *single cell* design as most of the information is shared between the different locations. A multiple cell design, with one cell in each main location, would create much administration overhead to keep all the locations current.

An industrial company differs from the above by the specialization of its locations, such as production, research, headquarters, or sales. In this case, there will be a very high volume of local data flow within departments compared to the cross department data flow.

For this company profile *(organization type 2)* we would recommend *multiple cells*, one cell being designed for each of the most “independent” departments.

In general, the more the departments are independent in the company, the more they are good candidates to have their own cell configuration. This means they also need resources and skills to perform cell administration. Such independence may be the result of varying missions, the mission of the departments, or from geography. An international company may have locations in countries where regulation and infrastructures are so specific that they can′t fit in a single image configuration.

Multiple cell design should also fit very well to a structure such as a university *(organization type 3)* where all the departments have independent goals; they would run separate cells.

2.4.2 Existing Network and Systems

The cell design will have to show how to reuse, replace, add, or migrate existing physical components, among which are:

- The network: a sizing of the traffic generated by DCE will allow assignment of line usage in the cell considering line speeds. The designer will avoid using slow lines for heavy traffic (frequent authentication, DFS file location process, and so forth). He will also add new lines to increase availability to access critical servers within cells.
- Installed systems: the planner will have to check if the systems in place support DCE functions and, if not, which systems need to be added or replaced in the configuration. DCE requires additional computing and storage resources, which are described in the *Configuring and Getting Started* guide of each platform.

This leads us to look in more detail at the network impact on cell design.

2.5 Network and Systems Layout

DCE generates types of network traffic that should be considered when fitting the cell design with the network configuration.

2.5.1 DCE Network Traffic

DCE network traffic comes from the communication process between the DCE components, and also from the DCE applications themselves.

- CDS Network Traffic: comes from the “advertising” mechanism of the CDS, the “soliciting” traffic of CDS clerks, and the “skulking” traffic by which CDS updates its replicas (master and read-only). The skulk can occur in three different ways:
  - By a command of the administrator (`set directory /.:/dirname to skulk`)
  - Following a management activity (any creation/delete/modify action of the administrator on replicas or directories)
  - Automatically; skulk starts at a routine time interval (the *background skulk time*)
- Security Server Network Traffic: is generated between the master and the slave. (Note that the security master server is the single DCE server facility that is allowed to update the security data found in the registry; the slave

servers are allowed to read from the registry, but writing is only allowed in the case where a master update is to be reflected). Each time an update occurs on the master registry, this update is propagated to the slave registries immediately. The traffic is manageable since it results only from voluntary updates to the master database.

- DTS Traffic: is mainly synchronization messages in a cell between the DTS servers. This traffic can be contained within a physical LAN by declaring the DTS global server as “Non courier.” This way the global DTS will not look for other global servers in foreign LANs for the purpose of clock synchronization.
- RPC Traffic: is the traffic created to perform the actual work of the application. The designer should recognize there will be some overhead for general RPC handling.
- Login Traffic: is necessary in order to get a ticket from the authentication service and a Privilege Attribute Certificate from the privilege service of the security server.
- DFS Traffic: performs name resolution. Traffic exists to deliver pieces of DFS files to clients, although this is generally lower when compared to NFS traffic to perform the same function. There is also traffic between the *File location servers* as they determine which of them is the master.

All this traffic is handled by a LAN network without major restrictions.

2.5.2 Network Configurations

A single cell, except for test purposes, is generally built on top of several LAN networks. Interconnections of these LANs is critical for the cell design.

- If the LANs are connected through fast multi-protocol routers and through a high speed back-bone (FDDI or ATM), they can be considered as a single LAN. The design of the cell has to be balanced for reliability and performance as if it were a single LAN.
- If the LANs are connected through low speed links, the cell designer will try to reduce the traffic through these connections. Some recommendations in this area are:
  - Locate the application server on the same LAN as most of the users of the application.
  - If a CDS replica is on a remote site connected by slow links, the administrator will check to have local directories replicated on this remote site.
  - The lifetime of tickets delivered by a security server should be long enough to reduce login overhead.
  - FLDB must not be installed at the DFS replica location over slow links (high traffic deteriorates cell response times). Then to reduce the access to FLDB on the main site, DFS has to be organized so that the hierarchy at mount points is as flat as possible to reduce path name resolution process
  - An alternate link has to be installed to increase the availability of the cell.

A multiple cell will be the combination of the two cases above (single LAN or “single LAN-like” cells and multiple LAN cells). The same restrictions apply with additional considerations for cross-cell communications:

- GDS (or DNS) inter-cell name resolution process
- Security server inter-cell trust relationship
- DFS ACL definitions

Servers locations are discussed in more detail in Chapter 3, “Sample Scenarios - Theoretical and Real” on page 23.

2.6 Administration Policies and Tools

Each IT installation has its own policies ruling the operations:

- Naming conventions
- Security rules
- Files organization

As well, administration tools are used to manage operations:

- Security administration
- Network management
- System management
- File management

The DCE cell designer will have to review how DCE administration tasks can be integrated in the existing procedures and what additional tools are required.

Cross-cell coordination activities will be required in a multiple cell environment. In general, multiple cells will generate additional administration tasks; the basis of these tasks will be a well-structured organization.

2.6.1 Naming Conventions

Cell naming is an important part of the planning process for several reasons:

- The name is an identifier that will be used to communicate with other cells. Even if you plan a single cell design, your cell may have to communicate with foreign cells in the future. Your own organization can grow and you will have to decide to add a cell.
- The cell name is the basis for the authentication process in the cell.
- The name has to comply with standards to be registered in DNS or GDS and so to be unique for this cell identification. The registration process has to be completed by the naming authorities before you start configuring your cell. For more details see *OS/390 OpenEdition DCE Planning*, SC28-1582 and *OS/390 OpenEdition DCE Administration Guide*, SC28-1584.

In the global name, a part of the cell name remains the choice of your organization. This part can comply with your internal naming conventions or agreements made within your organization. By maintaining this structure, the name chosen will not be likely to require a change in the future.

Changing the name of a cell can be rather difficult once the configuration is complete, as the name appears in numerous definition parameters. A change would normally be accomplished by a complete unconfiguration of the cell followed by a new configuration.

2.6.2 Security

The two main aspects of security are organization and administration: organization to grant authorizations and privileges to the people according to their role in the company, and administration to insure that the security policies are applied.

2.6.2.1 Security Organization

Security rules have to be implemented in the DCE configuration:

- In the registry data base that manages cell security: the entries in this database are entities involved in the operation of the cell: users and servers. These entries are called *principals*. Principals are gathered in *groups*, themselves gathered in *organizations*. The collection of a principal + a group + an organization is called an *account*. See also 4.6, " Multiple Cell Definitions" on page 67, and 4.6.3, "Security" on page 69.
- In the authentication service which issues tickets used by principals to access remote services. Ticket lifetime is an important parameter:
  - If it is too short, there will be frequent logins to the authentication service, which implies traffic overhead (in a cross cell authentication process this can also have a strong impact on performance).
  - If it is too long, the risk of intruder actions into the cell increases.
- In the ACL facility, to establish and grant access to resources.

The cell designer will work here with the security administrator to map the existing security administration to the registry database organization. Administrative policies are also registered in the registry database.

2.6.2.2 RACF/DCE Interoperation

When an OS/390 system is member of the cell, a security policy is already in place on this system through RACF or a security program using the SAF interface. The decisions in the OS/390 security policy may be extended to the DCE environment in either of two ways:

- The DCE security server can be installed on the OS/390 system; the registry database becomes a translation of RACF facilities.
- RACF/DCE interoperation utilities can be used by the administrator:

  To register DCE users in the RACF database
  To register OS/390 users in the DCE registry database
  To link both databases

By this means, a consistent security policy can be implemented across the platforms, using the existing security organization. For more details, see *OS/390 OpenEdition DCE Security Server Overview*, GC28-1938.

2.6.2.3 Security Administration

A focal point to manage the registration of all the users may be a solution to the security administration of complex configurations. A family of IBM products has been announced to implement this type of security function: the *Distributed Security Manager* family.

- *DSM/AIX* gives to the security administrator a business-oriented interface for administering users and resources on all managed security systems: *any DCE compliant platform*. The product introduces role-based security administration, which means the security administrator authorizes access to

resources according to people's functions in the organization. Access rights to resources are stored in a DSM owned database. Auditors can conduct preventive security checks before any security violations might be attempted. Offers request queuing to manage remote security systems, regardless of their server availability.

Agents (sets of functions that link DSM with a security system to be managed) can be developed using interfaces provided by DSM. DSM/AIX requires and uses a full DCE infrastructure.

DSM/AIX has been available as a beta program since December 1995. In this program, the client runs on OS/2 Warp, the server on AIX and registers its management rules in a DB2 for AIX database. See Figure 4.

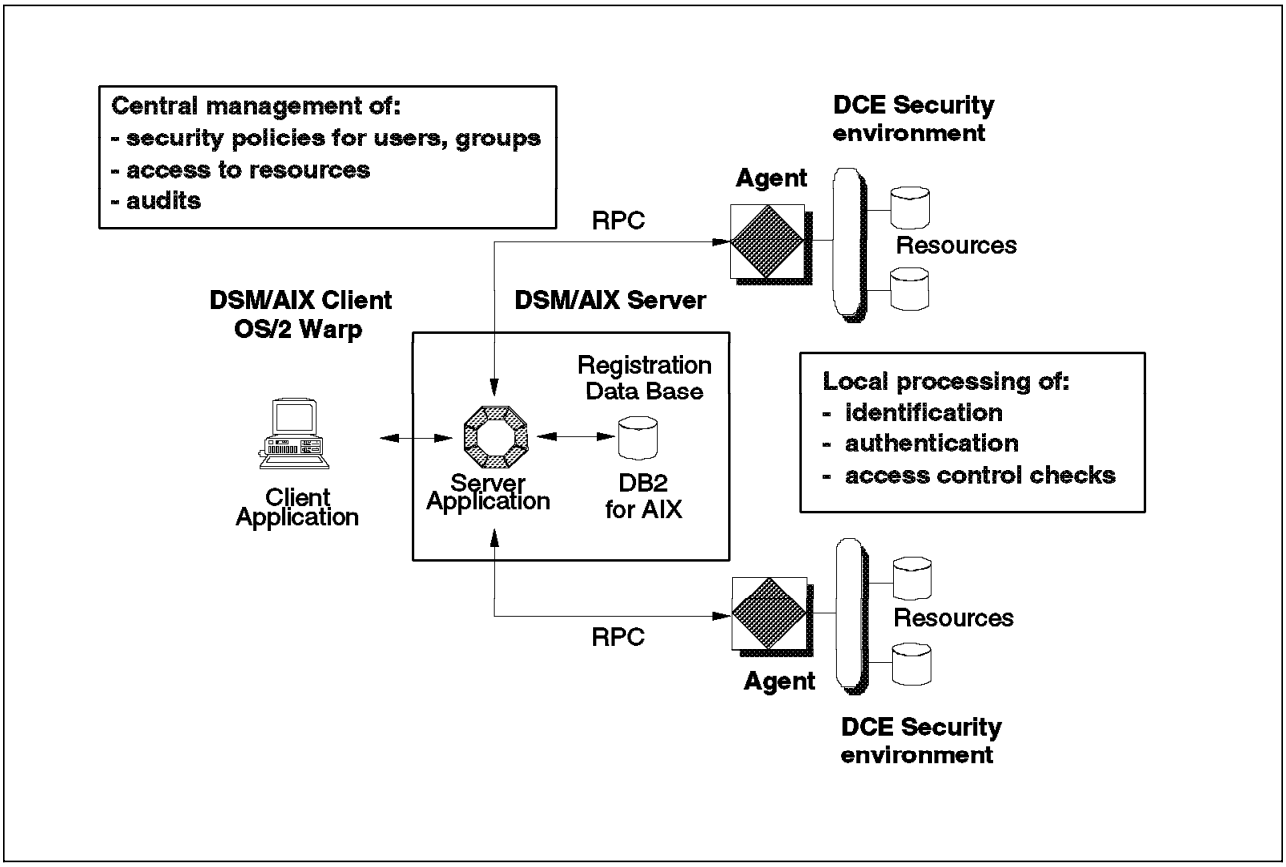


Figure 4. DSM/AIX Beta Program Principles

- *DSM/MVS* provides consistent security administration across multiple heterogeneous platforms including MVS, VM, OS/400, OS/2, and Novell Netware. An agent is provided for each of these platforms. The administrator uses a GUI interface on OS/2 to access these environments. The availability of a DCE agent for DSM/MVS is a strong customer requirement.

In a multiple cell environment, this kind of tool is required. It keeps the local security administrator in charge of the cell and also gives to a security supervisor the means to check and apply global security. Each user receives consistent rights according to his role in the enterprise.

2.6.3 File Management

DFS is an application built over DCE, which allows users to access and share files stored on a file server anywhere in the cell. Since the naming of the files is unique in DFS, cross-cell access and sharing of the files is possible without additional definitions. The administrator will only have to check the ACL authorizations. See 4.6, " Multiple Cell Definitions" on page 67.

OS/390 OpenEdition supports a file system called *Hierarchical File System (HFS)*, which is in bytestream format (data encoding could be EBCDIC or ASCII, but usually is ASCII). OS/390 also supports DFS (level 1.0.3a of OSF/DCE). The DFS file server runs on OS/390 but the fileset location server must be elsewhere in the cell (on an AIX system, for example).

MVS data (sequential or PDS files) can be copied to or from an HFS. HFS can be exported to the DFS file system. This is a way to make MVS data available to the DCE users. See Figure 5. HFS is managed through DFSMS.

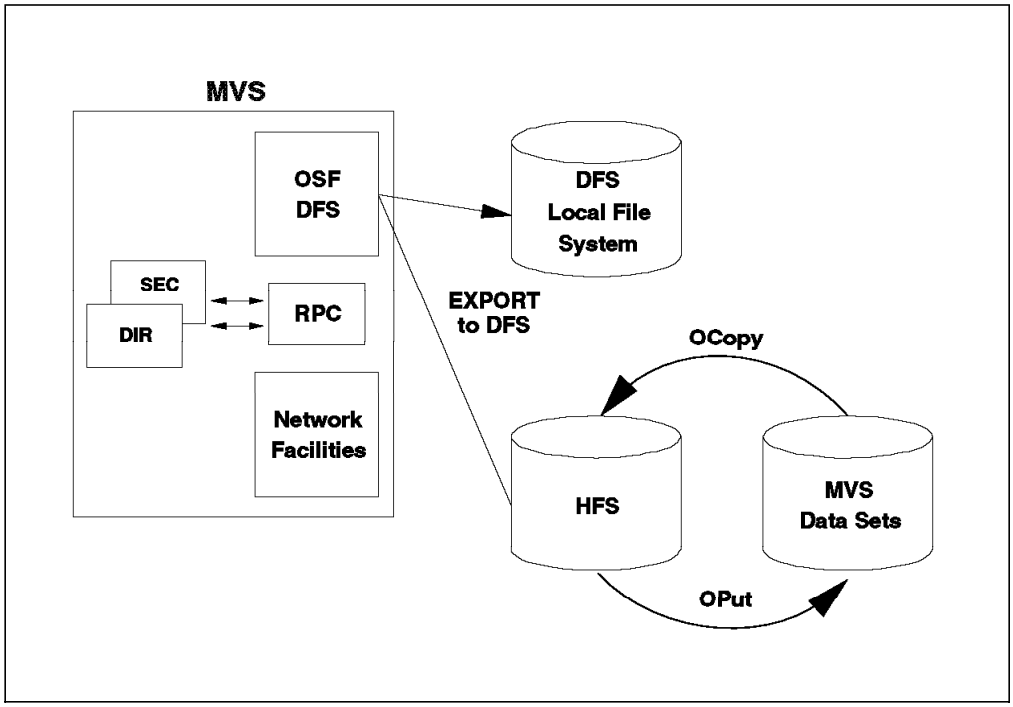


Figure 5. DFS on MVS (OS/390) and HFS

For more information, refer to *OS/390 OpenEdition DCE Distributed File Services, Configuration and Release Notes*, SC24-5723 and *OS/390 OpenEdition DCE Distributed File Services, Administration Guide*, SC24-5722.

2.6.4 Cell Administration Summary

DCE cell administration requires several different roles and skills to perform all the tasks. A multiple cell implementation implies each cell is capable of performing its administrative functions. Then, if the cells are really independent, minimal or no additional administration is required (minimal is GDS or DNS registration). If the cells need a strong consistency between them to operate (data and process sharing), a cross-cell administration has to be organized and performed.

2.7 Cost Element

The budget of the project gives the general scope in which the cell designer has to make his decisions. This budget must include:

- The cost of development of the new applications
- The cost of implementation of the new applications
  - Network, hardware and software installation
  - Application functions deployment
  - Technical education of the IT personnel
  - End-user education
- The cost of the management system of the new applications

To fit into the budget, the designer must sometimes choose a different technical implementation of what he originally planned. If this change has an impact on the service level of the application (performance or availability), it must appear in a *service level agreement* between the IT department and the end-users department.

The different scenarios in Chapter 3, “Sample Scenarios - Theoretical and Real” on page 23 show how the implementation and the administration differ between single cell and multiple cells design. The cost of these differences is always an important element of choice of the design.

Chapter 3. Sample Scenarios - Theoretical and Real

In this chapter, we start from two different samples for network and system lay-out (cases 1 and 2), and discuss the consequences of a single cell design (scenarios 1A and 2A) and of a multiple cells design (scenarios 1B and 2B). Comparison tables summarize the differences.

Note

The recommendations for DCE servers implementation come from the interpretation of studies reported in previous redbooks and DCE documents. They are not the result of physical implementations, measurements or tests.

3.1 Scenario 1 - DCE Cell over a Single LAN Structure

Here, the definition of “single LAN structure” is: one or more LANs interconnected through high-speed connections.
This means that there are no slow links in the configuration.

Since no limitations will exist from the network point of view, the implementation will be driven mainly by organizational considerations. In this section, we provide the main guidelines for determining the placement of the servers.

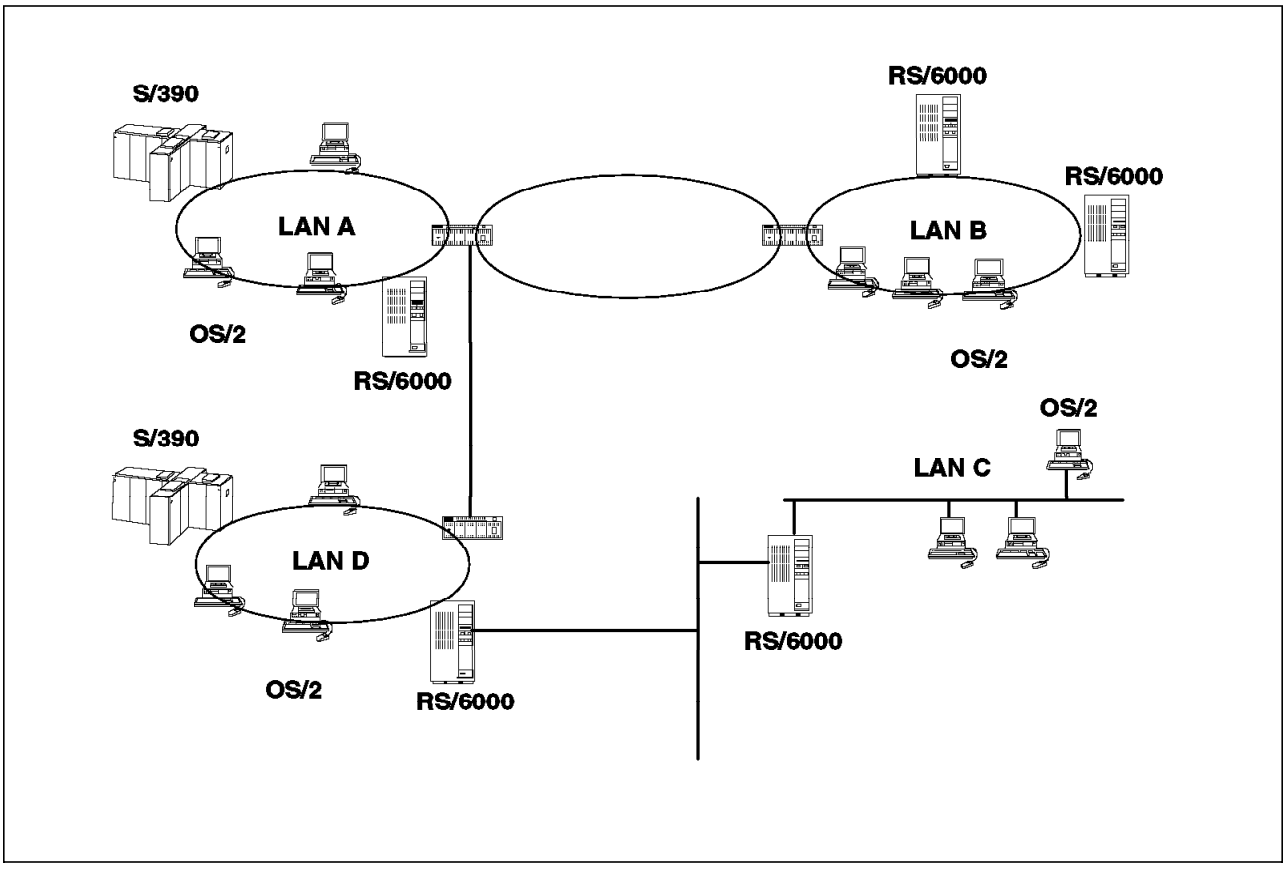


Figure 6. Network and Systems Layout. Infrastructure can be considered as a Single LAN.

3.1.1 Network and Systems Layout

3.1.1.1 Network

The network is organized on four operational LANs called A, B, C, and D. Operational means LANs with servers and users. The other LANs on Figure 6 on page 23 are only interconnection LANs.

- A and B are connected through high speed routers on a token ring LAN (not designated as an operational LAN).
- A and D are connected through high speed routers with a direct high speed link.
- C and D are connected through RS/6000 systems acting as routers on an ethernet LAN (not designated as an operational LAN).

3.1.1.2 Systems

- One S/390 system is on LAN A and one on LAN D.
- There is an RS/6000 System on each of LANs A, B, and D; there are two RS/6000 Systems located on LAN C. The main DP center location is LAN A; on LAN D, a S/390 system is managed from LAN A.
- Application servers are running on each LAN. Two application servers are running on LANs A and D (one on S/390 and one on RS/6000).
- LANs A and D have the largest number of users, LAN B has fewer users and LAN C has the smallest number of users.

Note

OS/2 WARP server can perform DCE security server and directory server roles. To simplify the scenarios, DCE servers are set only on S/390 or RS/6000 systems. Keeping in mind functional limitations (see 4.4, “Server Configurations” on page 50) and possible performance limitations, a RS/6000 server can be replaced by an OS/2 WARP server for any supported DCE service.

3.1.2 Scenario 1A -Single Cell Design

We review each DCE component and describe a configuration for the DCE servers that takes into account the network and systems layout. The location of the main DCE servers is shown in Figure 7 on page 25.

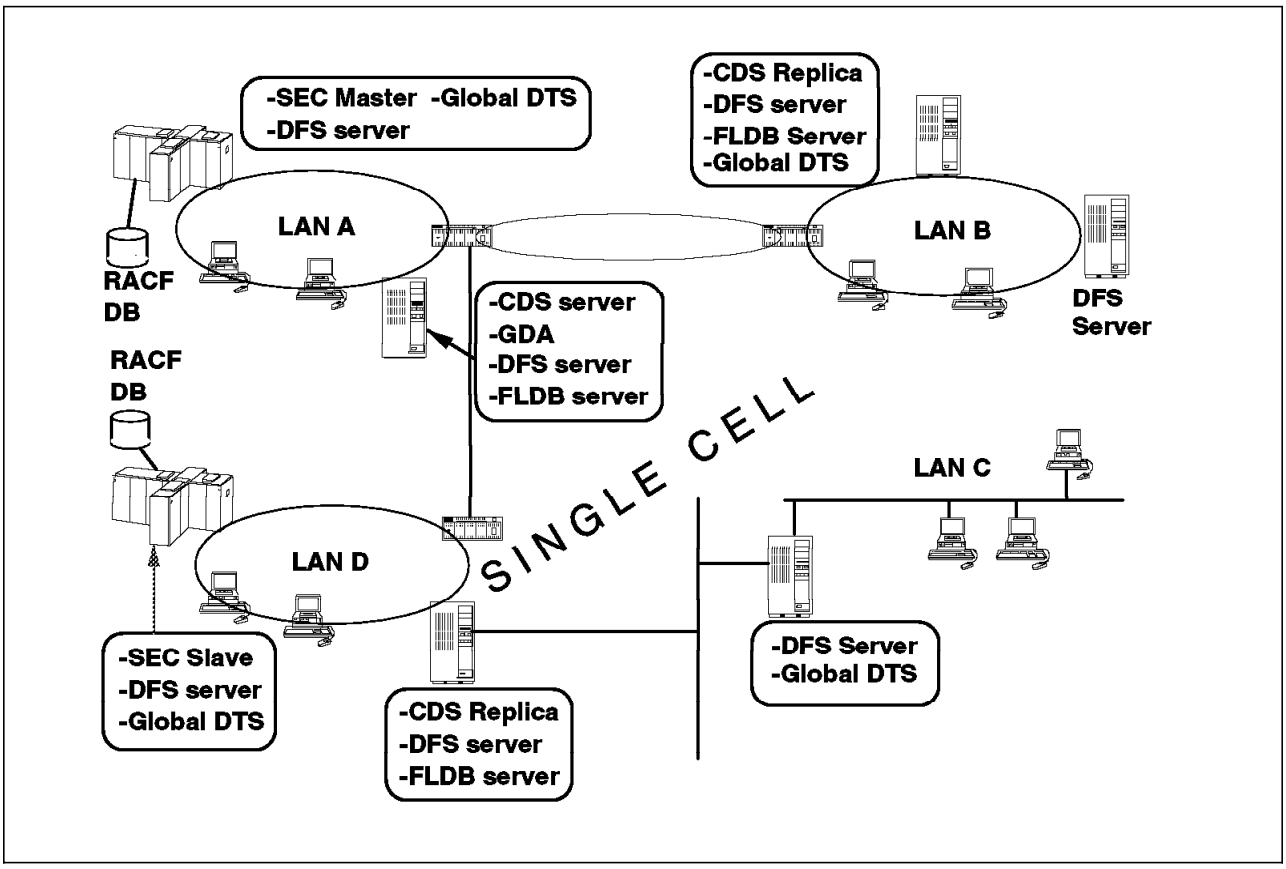


Figure 7. Scenario 1A - DCE Servers Layout. Single Cell Design over a LAN Structure.

3.1.2.1 Directory

The directory server is installed on LAN A (RS/6000). Replicas can be installed on LAN B (RS/6000) and LAN D (RS/6000) for performance and availability reasons.

HACMP supports DCE applications. For availability reasons only, an HACMP cluster may be considered instead of CDS replicas. In this situation performance has to be checked as all directory searches will arrive at a single point of the cell.

GDA is active on RS/6000 on LAN A and cell naming complies to standards (DNS or GDS) so communications with foreign cells may be set up if needed.

3.1.2.2 Security

The security server is installed on S/390 on LAN A, taking advantage of RACF/DCE interoperation utilities. The S/390 on LAN A also manages MVS users of LAN D through RACF distributed database functions. A slave security server can be installed on the S/390 system on LAN D.

3.1.2.3 DFS

A DFS server can be installed on each S/390 system. OS/390 DCE is not at level OSF/DCE 1.1, so a fileset location database (FLDB) server must be installed on an AIX system on LANs A and D. To optimize the voting process of DFS a third FLDB server is installed on an RS/6000 system on LAN B.

3.1.2.4 Time Server

A global time server is required at the LAN level. This component runs on OS/390, AIX or OS/2. If an S/390 system is part of a sysplex, the sysplex timer can be used as an external time provider.

3.1.2.5 Administration

Cell administration is an extension of the existing management performed from LAN A:

- CDS management is a new discipline which must be created.
- Security management can be integrated to RACF administration but requires new skills to perform DCE registry tasks and ACL tasks. Through *dcecp* commands and scripts, every DCE system can be administered.

  The master security server and the slave security server are on the same platform (OS/390) to keep a consistent access to security functions from the administration point of view. It should be noted that this is not a major advantage; the function can be performed through the same DCE commands, either through SMIT or ISPF.
- DFS permissions (ACLs) are related to the security tasks. DFS definitions require new file management skills to establish which data will be part of the filesets, to manage the filesets, and to control backup functions.

3.1.3 Scenario 1B - Multiple Cell Design

On the same network structure, the DCE design is made of three different cells:

- We will assume that LAN A is a legacy system and is defined as *cell 1*.
- LAN B is a department previously running on LAN A systems and now running on its own system on LAN B; it is defined as *cell 2*.
- LAN C + LAN D is the DP system of a company joining the “legacy” company; it is defined as *cell 3*.

See Figure 8 on page 27.

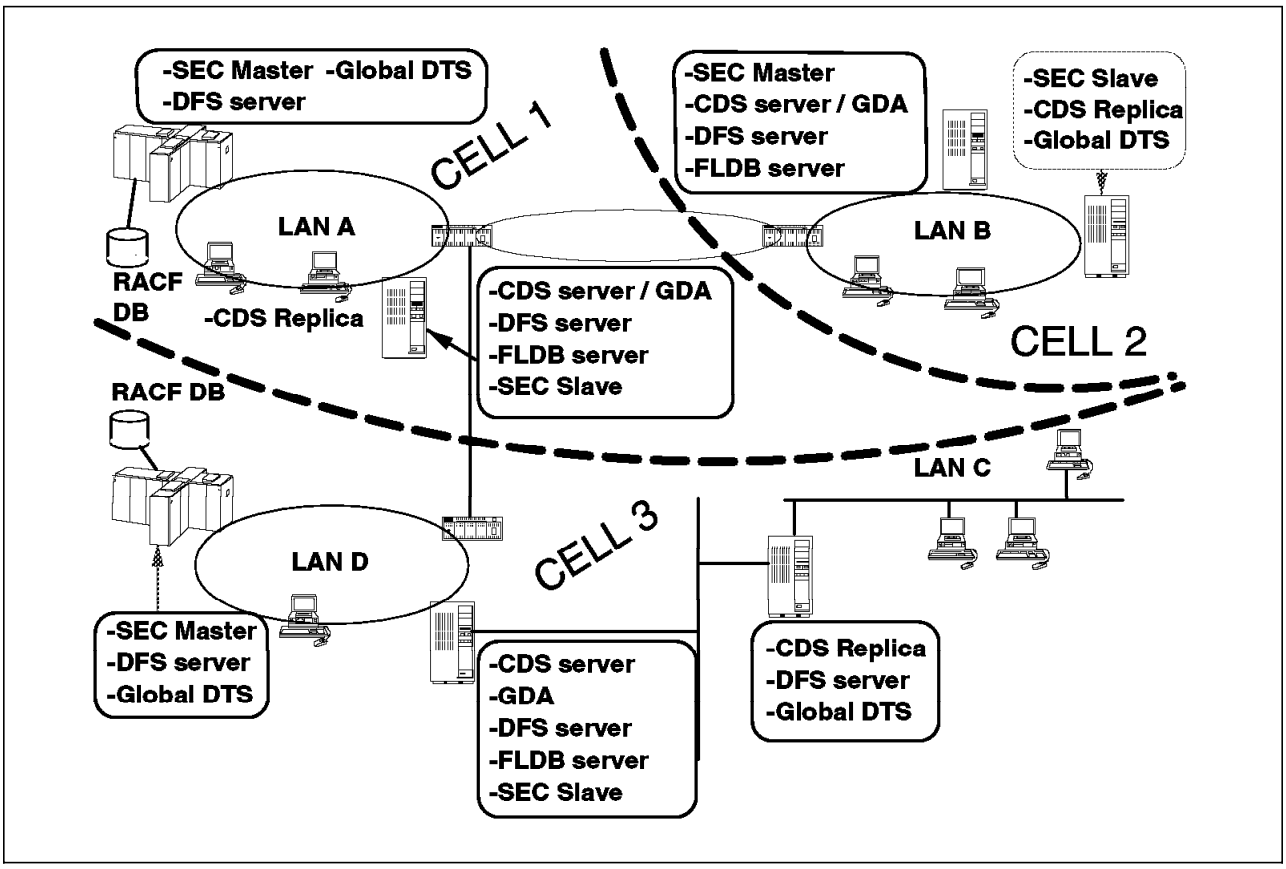


Figure 8. Scenario 1B DCE Servers Layout. Multiple Cells Design over a LAN Structure.

3.1.3.1 Directory

A directory server must be installed in each cell. Each of these servers has also GDA active and standard cell naming. For availability reasons, a replica of each of these directory servers can be installed in each cell.

3.1.3.2 Security

In cells 1 and 3, security servers run on S/390 systems and interoperate with RACF. Slave security servers can be installed on RS/6000 systems of the same cells.

3.1.3.3 DFS

The FLDB voting process recommends at least three FLDB servers in each cell. They have to be installed on RS/6000 systems. Two additional FLDBs within each cell should be considered for production environments.

3.1.3.4 Time Server

A global time server is required at the LAN level, as in scenario 1A.

3.1.3.5 Administration

The administration considerations that arise for scenario 1A are still valid as far as cell 1 and cell 3 are concerned. These two cells may extend their S/390 administration to DCE with additional skills as described in 3.1.2.4, “Time Server” on page 26.

Cell 2 has to build an administration on RS/6000.

If the three cells have to interoperate closely, that is, if users of any of the cells has to access applications on one of the two other cells, a cross-cell administration role must be defined. Interoperation implies one of the following:

- Deciding on a point of control for the three cells from which cross-cell definitions will be established
- Naming a coordination group between the members of the administration group of each cell

3.1.4 Scenarios 1A and 1B Summary

We show in the table a comparison of the two scenarios (number of servers and administration considerations).

Table 2. Scenarios 1A and 1B Comparison

Discipline	Scenario 1A single cell	Scenario 1B multiple cells
Security	S/390-A Master, S/390-B Slave	S/390-A Master, RS/6000-A Slave RS/6000-B Master, RS/6000-B #2 Slave S/390-D Master, RS/6000-D Slave
Directory	RS/6000-A Master, RS/6000-B Replica, RS/6000-D Replica GDA defined on RS/6000-A	RS/6000-A Master, RS/6000-A #2 Replica *(additional or OS/2)* RS/6000-B Master, RS/6000-B #2 Replica RS/6000-D Master, RS/6000-C Replica GDA active on each CDS Master machine
DFS	All S/390 and RS/6000 in the cell are DFS servers: Single “file space” in the cell RS/6000-A FLDB server, RS/6000-B FLDB server, RS/6000-D FLDB server 3xFLDB′s in the cell for voting process, availability and performance.	In each cell, S/390 and/or RS/6000 are file servers RS/6000-A FLDB server, RS/6000-B FLDB server, RS/6000-D FLDB server; 2 additional FLDB servers per cell have to be considered in a production design.
Time	S/390-A Global DTS, RS/6000-B Global DTS, S/390-D Global DTS, RS/6000-C Global DTS One global DTS for each physical LAN	S/390-A Global DTS, RS/6000-B Global DTS, S/390-D Global DTS, RS/6000-C Global DTS One global DTS for each physical LAN.
Administration	One DCE SEC/RACF administration for the whole cell, managed from S/390 on LAN A. CDS managed from S/390 on LAN A through dcecp commands. DFS managed from RS/6000 on LAN A.	• DCE SEC/RACF administration on S/390 on LAN A • CDS managed from S/390 on LAN A through dcecp commands • DFS managed from RS/6000 on LAN A • *Cross-cell administration* • DCE administration on RS/6000 LAN B • DCE SEC/RACF administration on S/390 on LAN D • CDS managed from S/390 on LAN D through dcecp commands • DFS managed from RS/6000 on LAN D.

It appears that for this kind of LAN network:

- Multiple cells design requires more machines and an additional administration organization.
- Multiple cells may have better performance only if applications and their users are in the same cell.

- Single cell design is more flexible for servers and users location that is application servers and their users may be located anywhere in the cell.
- Multiple cell design should be preferred only for organization reasons see 2.4.1, “Types of Organization” on page 14 (independence of the department using and managing its own cell).

3.2 Scenario 2 - DCE Cell over a WAN Structure

In this topic, we start from the same basic LANs and connect them through slow links in order to create a WAN.

Now we have some constraints in the network, and we discuss how they change the cell design.

We use the same WAN for a single cell design and a multiple cell design.

3.2.1 Network and Systems Layout

3.2.1.1 Network

The network is organized on four operational LANs called A, B, C, and D. See Figure 9 on page 31.

- LANs A and B and LANs A and D are connected through low speed dedicated links.
- LANs C and D are connected through RS/6000 systems acting as routers on an X25 network.

3.2.1.2 Systems

- One S/390 system is on LAN A and one is on LAN D.
- LANs A, C, and D each contain a RS/6000 system; LAN B contains two RS/6000 systems. The main DP center location is LAN A; on LAN D, the S/390 system is piloted from LAN A.
- Application servers are running on each LAN; two application servers are running on each of LANs A and D (one on S/390 and one on RS/6000).
- LANs A and D have the largest number of users, LAN B has fewer users and LAN C has the smallest number of users.

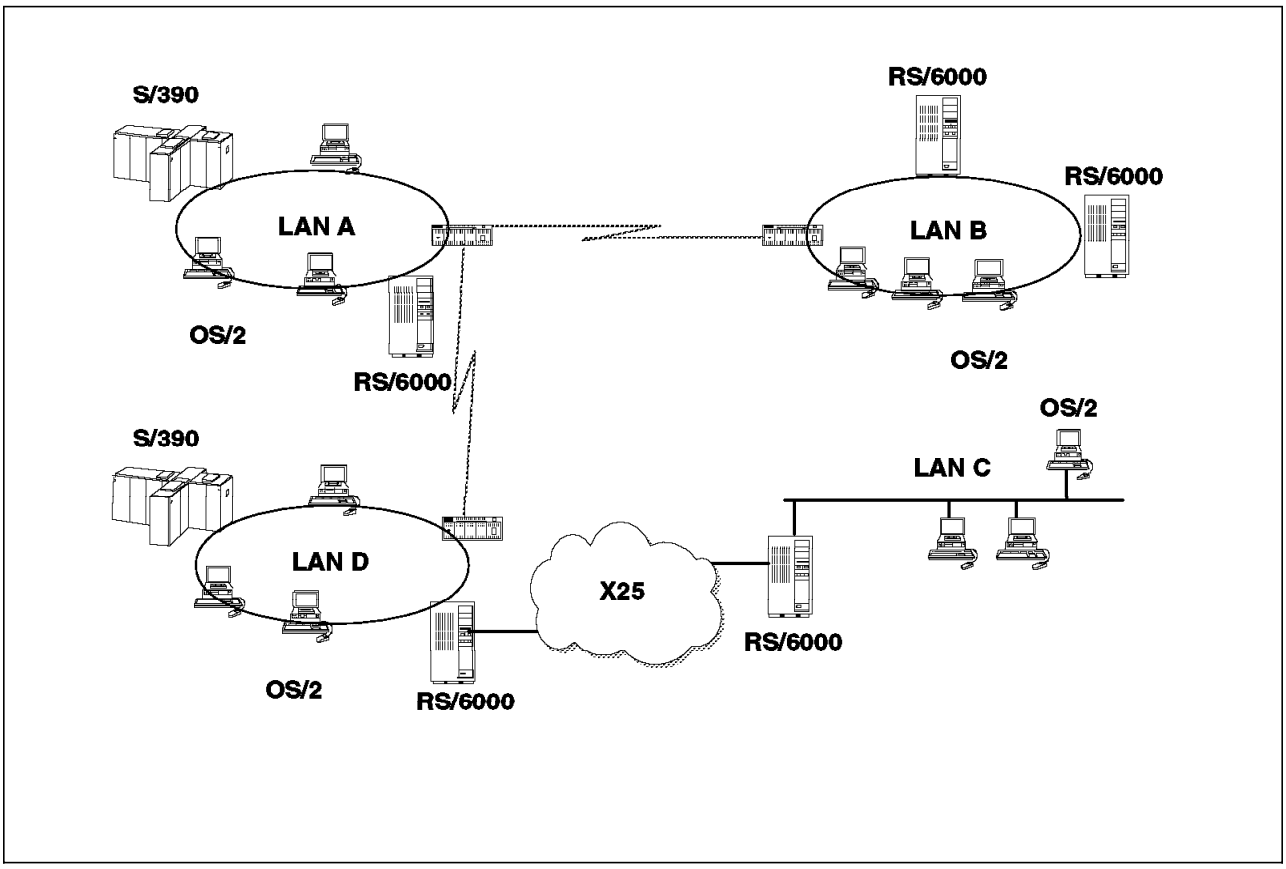


Figure 9. Network and Systems Layout. WAN Infrastructure

3.2.2 Scenario 2A - Single Cell Design

The main guideline for the design of the cell in this case is to implement DCE servers in such a way as to minimize the network traffic generated by DCE functions over the slow links. The location of the main DCE servers is represented in Figure 10 on page 32.

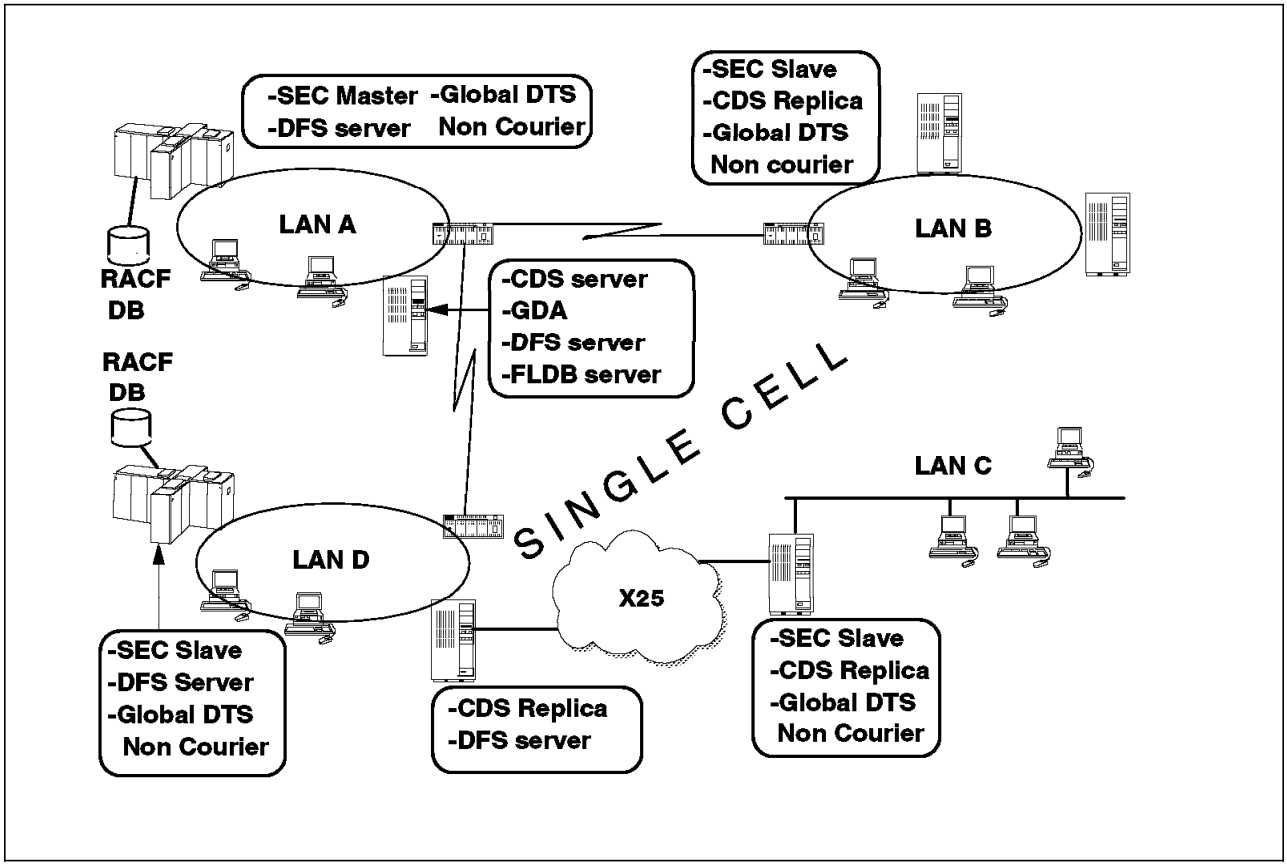


Figure 10. Scenario 2A - DCE Servers Layout. Single Cell Design - WAN Structure.

3.2.2.1 Directory

The directory server is installed on LAN A (RS/6000). We recommend to replicate the local directories of the CDS related to each LAN, on the local server of this LAN. So we will have replicas on LAN B (RS/6000), LAN C (RS/6000) and LAN D (RS/6000) with local directories only. In this way, the first search in the CDS will most of the time remain on the local LAN.

The cell administrator has defined a skulking interval long enough to prevent skulking traffic during TP hours. This defines the currency of the replicas to be 24 hours.

The cell administrator must also avoid maintenance operations on the CDS components:

- Adding or removing a replica
- Creating or deleting a directory

because these operations generate skulking.

GDA is active on RS/6000 on LAN A and cell naming conforms to standards (DNS or GDS) so communications with foreign cells may be implemented.

3.2.2.2 Security

The security server is installed on S/390 on LAN A, taking advantage of RACF/DCE interoperation utilities. The S/390 on LAN A also manages MVS users of LAN D through RACF′s distributed database functions. Slave security servers are installed on each LAN, RS/6000 on LAN B, RS/6000 on LAN C and S/390 system on LAN D.

Authentications, tickets and privileges can be delivered locally, that is, at the LAN level. Once the slave security server has been created, it is accessed locally in a read-only mode. Traffic is generated between the master server and the slave server only in case of update on the master. Note, however that the way to ensure local usage is through a variable (PE_SITE), but usage of PE_SITE implies that logins will fail if local services are not available, so this method of guaranteeing usage of the local server is not recommended.

In the implementation we recommend above, the technical DCE traffic is minimal when application servers are also used locally. If a user of LAN B works mainly with an application server on LAN D, performance problems can be encountered. In this case, client and server localization has to be optimized at the LAN level.

3.2.2.3 DFS

DFS implementation is difficult on such a network layout. As FLDBs exchange heavy technical traffic between them, it is not recommended to install second and third FLDBs on remote locations connected through slow links. In scenario 2A, DFS is only configured for LAN A and LAN B users because it is the largest population. S/390 and RS/6000 are file servers on LAN A and RS/6000 is also FLDB server. S/390 and RS/6000 are file servers on LAN B but there is no FLDB on LAN B.

As we assumed that these two cells run four application servers, we can encounter performance problems with only one FLDB.

At the opposite of scenario 1A, there is no common file space for all the users of this single cell.

3.2.2.4 DTS

A global time server is defined at the LAN level as in scenario 1A and 1B. In this case, it is defined as *Non Courier* so the time server will not try to synchronize with the other global time servers in the other LANs.

3.2.2.5 Administration

Cell administration is similar to scenario 1A. Administrators will have to perform administrative tasks that can generate traffic between the LANs (CDS modifications, Security server replication and so forth) outside of TP opening hours. The network and system load generated by these tasks has to be evaluated. The administrators must check to be certain the elapsed time required is consistent with the time windows available.

Note

The availability of the cell depends mainly on its weak points which are the slow network links. Alternate network routes and links have to added to this design to bring the cell at a higher level of availability. In this way, DCE and applications servers have a higher probability to be accessed even if a part of the network fails.

3.2.3 Scenario 2B - Multiple Cell Design

This case is similar to scenario 1B. The servers can be implemented in the same way. Each cell provides its own independent DCE functions. This guarantees a good level of service within each cell.

The only difference is to define the Global DTS as *Non Courier* in cell 3 on LAN C and LAN D.

As in scenario 1B, there is no FLDB on LAN C.

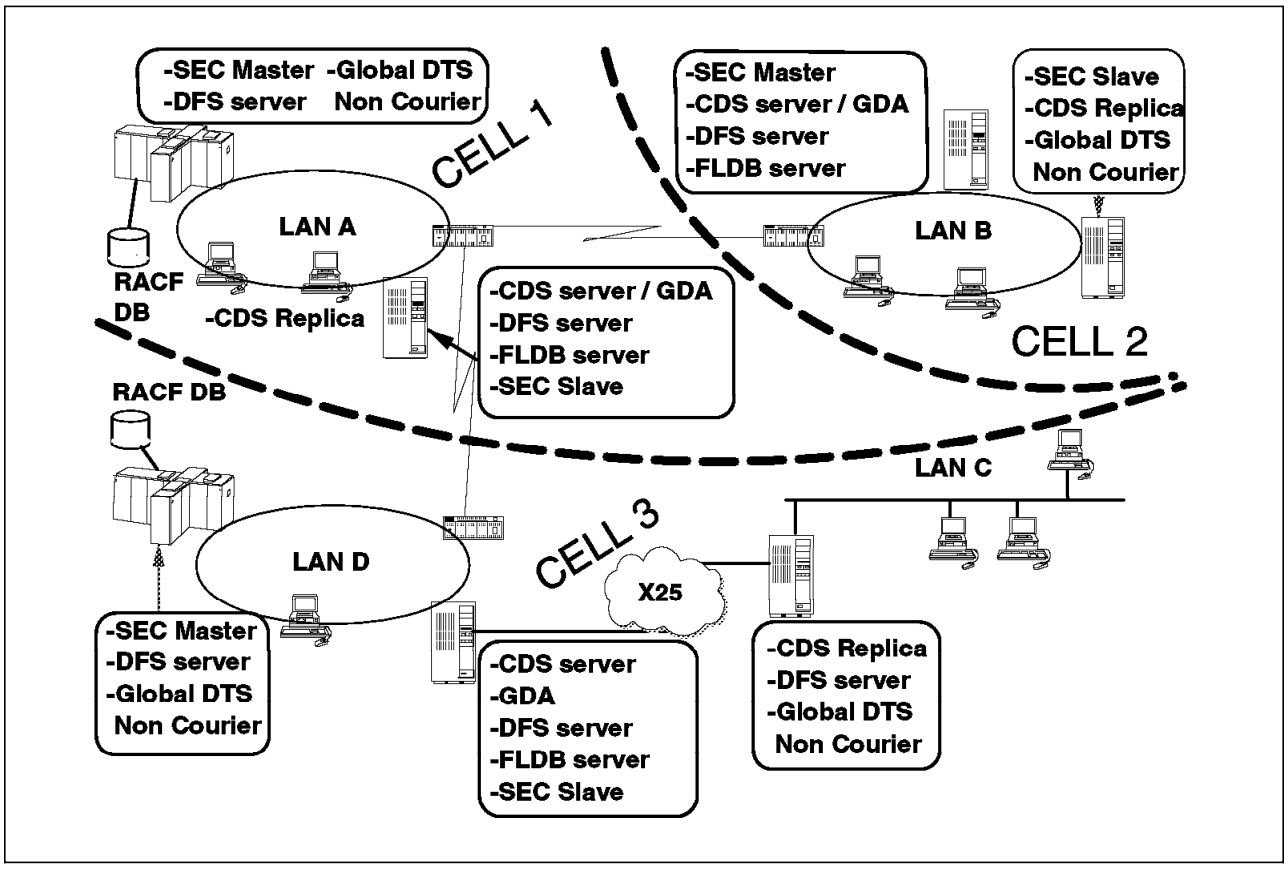


Figure 11. Scenario 2B - DCE Servers Layout. Multiple Cell Design - WAN Structure.

3.2.4 Scenarios 2A and 2B Summary

We show in the table a comparison of the two scenarios (number of servers and administration considerations).

Table 3. Scenarios 2A and 2B Comparison

Discipline	Scenario 2A single cell	Scenario 2B multiple cell
Security	S/390-A Master, S/390-D, RS/6000-B, and RS/6000-C Slaves. To prioritize local security functions (authentication, privilege), manual configuration of *pe_site* for clients is required.	S/390-A Master, RS/6000-A Slave. RS/6000-B Master, RS/6000-B #2 Slave. S/390-D Master, RS/6000-D Slave.
Directory	RS/6000-A Master, RS/6000-B, RS/6000-C, and RS/6000-D Replicas. Only local directories are selected for replicas.	RS/6000-A Master, RS/6000-A #2 Replica *(additional or OS/2)*. RS/6000-B Master, RS/6000-B #2 Replica. RS/6000-D Master, RS/6000-C Replica.
DFS	OS/390-A, OS/390-D, RS/6000-A, RS/6000-D File servers. RS/6000-A FLDB server. Only one FLDB in the cell; performance issue.	In each cell, OS/390 and or RS/6000 File servers. RS/6000-A FLDB server, RS/6000-B FLDB server, RS/6000-D FLDB server. 2 additional FLDB servers per cell have to be considered in a production design.
Time	S/390-A Global DTS, RS/6000-B Global DTS, S/390-D Global DTS, RS/6000-C Global DTS. One global DTS for each physical LAN. All DTS defined as Non Courier.	S/390-A Global DTS, RS/6000-B Global DTS, S/390-D Global DTS, RS/6000-C Global DTS.
Administration	One DCE SEC/RACF administration managed from S/390 on LAN A. CDS managed from S/390 on LAN A through dcecp commands. Command delays may be observed over the network. DFS implementation limited to a part of the cell.	• DCE SEC/RACF administration on S/390 on LAN A. • CDS managed from S/390 on LAN A through dcecp commands. • DFS managed from RS/6000 on LAN A. • *Cross-cell administration*. • DCE administration on RS/6000 LAN B. • DCE SEC/RACF administration on S/390 on LAN D. • CDS managed from S/390 on LAN D through dcecp commands. • DFS managed from RS/6000 on LAN D.

In the case of this kind of WAN network:

- Multiple cells design is certainly the more efficient. It offers the best service within each cell.
- Single cell implementation requires more manual definitions (CDS directories choice, security server choice priority using *pe_site*,...).
- Single cell does not allow DFS implementation across all the LANs.

- Even *type 1* and *type 2* organizations (see 2.4.1, “Types of Organization” on page 14) would take advantage of a multiple cell design.

3.3 DCE Cell Real Implementation

Most of the implementations in production that we have heard of are structured on a single cell design. Other cells often exists in the organization but for test purposes only.

A typical implementation is:

- A DCE client application on Windows developed with GUI interface development tool. These tools (see 5.4, “DCE Development Tools” on page 76) provide interfaces to DCE objects and APIs to develop DCE applications.
- TCP/IP network or TCP/IP protocol over SNA using Anynet products.
- The second tier machine (AIX) may handle several roles:
  - CDS and DCE Security functions
  - DCE application server, (in this case the implementation is a 2-tier client/server application)
  - RPC to APPC conversions to send requests to a S/390 host
- The third tier machine, if there is one, is:
  - A UNIX machine.
  - A main frame (S/390 host), processing the calls from the client either in APPC, or in RPC using the Application support server functions for IMS or CICS (see 5.2.1.2, “DCE Application Support for MVS/ESA” on page 72).

    Note: Security server on OS/390 is only available since February 29th, 1996. With this new function, we can expect S/390 systems to become more and more second-tier machines with security and application roles.

Chapter 4. DCE Server Considerations

This chapter describes creating and configuring a DCE cell. Included are considerations on server and client components for the following services: Security services, Cell Directory Service (CDS), Distributed Time Service (DTS), Remote Procedure Call (RPC), Distributed File Service (DFS), and Global Directory Agent (GDA). It describes the differences while configuring DCE on the different platforms.

It also describes the relationships between the different servers when they are installed on different machines or platforms.

This chapter provides information on the following topics:

- Overview of configuration
- Initial cell configuration
- Further cell configuration
- Distributed File Service (DFS)
- Multiple cell definitions

For more details about installing and configuring DCE on the different platforms, as well as for more detailed descriptions about the DCE components, refer to the available system documentation. See also D.2, “International Technical Support Organization Publications” on page 101.

This chapter will not describe how to set up TCP/IP configurations, which is a prerequisite for DCE implementation.

4.1 Overview of Configuration

The configuration of a DCE cell occurs in two phases, which we can call “Initial Cell Configuration” and “Further Cell Configuration.”

In the first phase to initialize a DCE cell, the “Initial Cell Configuration,” configurations for the following components must be performed:

- Security Server
- CDS Server
- DTS Server

After the cell is up and running, you generally will not have to repeat any of these configuration tasks.

Phase two, the “Further Cell Configuration,” allows configuration of additional components (or their reconfiguration) into a cell:

- DCE and DFS clients
- Secondary CDS Servers
- Replica Security Servers
- Additional DTS Servers
- Audit Daemon
- Global Directory Agents
- DCE NFS to DFS Authentication Gateway
- DFS Servers

The configuration of these additional components is a task you can perform throughout the lifetime of the cell after initialization.

Before starting any configuration, remember the aspects of cell design shown in Chapter 3, “Sample Scenarios - Theoretical and Real” on page 23.

You may also use the worksheet on page 39 which will help you to define your cells:

Table 4. DCE Configuration Worksheet

	Enter Your Definitions
Cell Admin id	
Cell Admin Password	
Cell Name	
Name of Security Server Host	
IP-Addr of Security Server Host	
Name of CDS Server Host	
IP-Addr of CDS Server Host	
.	
.	
.	

			DCE					***DFS***					
Env	**TCP/IP Name**	**Internet Addr.**	**CL**	**CD**	**SS**	**DT**	**GA**	**CL**	**SC**	**FS**	**FD**	**BD**	**FR**
			X	X	X	X							
			X										
			X										
			X										
			X										
			X										
			X										
			X										
			X										
			X										
			X										
			X										
			X										
			X										

Where: CL=Client, CD=CDS Server, SS=Security Server, DT=DTS Server, GA=GDA, SC=System Control machine, FD=Fileset Database machine, FS=File Server machine, BD=Backup Database machine, FR=Fileset Replication Sever machine

4.2 Initial Cell Configuration

This section will show how to provide the minimum configurations for a DCE cell on the different IBM platforms. This is only possible on AIX, OS/2 WARP, and on OS/390.

Note: Initial cell configuration on OS/390 system is only possible in conjunction with an AIX system or OS/2 WARP system because DCE on OS/390 does not support CDS server functions.

On the other IBM operating system platforms (see also Table 1 on page 2) only DCE Client functions are supported.

4.2.1 OS/2 WARP Environment

Installing DCE for OS/2 requires OS/2 Version 3.0 or higher. The actual available DCE package for OS/2 is the OS/2 DCE 2.1 Beta level.

The DCE for OS/2 installation program (`INSTALL.EXE`) performs the following tasks:

- Creates the directory structures for each component
- Installs the appropriate files for each component
- Modifies the `CONFIG.SYS` file and sets the necessary environment variables
- Installs the DCE administration GUI
- Adds the configuration GUI

Entering `install` from the command line will display the installation window to guide you for the necessary steps. During the installation process, you will be prompted to select the DCE components, which can be:

- Client
- Administration GUI
- DFS Client
- Security Server
- Cell Directory Server
- Online Books
- Application Development Tools
- HPFS Include Files & Libraries
- Application Development Online Books
- HPFS Sample Programs

After Installation of the selected DCE components, a reboot of the system is required.

Also the configuration is icon/menu-driven using the GUI. Within different menus you have to enter the information to define an initial cell:

- Full, Administrative, Local configuration
- Cell Name
- Administrator User id and Password
- System Hostname an IP-Address

Also select:

- Server and client for registry service

- Initial server and client for Namespace service
- Local server for DTS service

The entered information can be stored in

`response file`

which you may select at the beginning of the configuration. When selecting *Run configuration*, the response is displayed in the *Configuration Progress* window. Informational, warning and error messages are logged in file `x:\opt\dcelocal\cfgdce.log`.

Notice

The installation of DCE for OS/2 requires about 56 MB of disk space without selecting online documentations and the application components.

However - about 50 to 60 MB of additional disk space (temporary) is required during the configuration phase for the initial cell.

When defining additional clients and servers within this cell - you should monitor the available disk space on your “master” system.

(For more details refer to the *IBM Distributed Computing Environment 2.1 for OS2: Beta II Getting Started.*)

4.2.2 OS/390 OE Environment

DCE has been provided on MVS since MVS 5.1.0. The latest version of DCE is shipped with OS/390 R1. OS/390 base features include:

- OpenEdition features, which are prerequisites to DCE:
  - OpenEdition MVS Services
  - OpenEdition MVS Debugger
  - OpenEdition MVS Shell and utilities
- DCE features
  - OpenEdition DCE Base Services (OSF/DCE level 1.1)
  - OpenEdition DCE Distributed File Service OSF/DCE level 1..0.3a)

OpenEdition DCE User Data Privacy is an optional feature of OS/390 for data encryption.

In the system, DCE brings a new address space *DCEKERN* (for DCE Kernel) in which the DCE daemons will run as individual processes:

- DCE daemon
- Security server daemon
- CDS Advertiser daemon
- CDS Clerk daemon

- DTS daemon (can be configured either as a server or a clerk)
- DTS Null Time Provider daemon
- Audit daemon
- Password Management daemon

Notice

Once the features of OpenEdition and DCE have been installed according to the *Program Directory for OS/390*, some prerequisites have to be verified before running the configuration:

- The CDS daemon must run on another machine in the cell as this server is not supported on OS/390.
- At least one DTS daemon must run as a server on another machine in the cell.
- The following address spaces are started on the system in this order:
  1. OpenEdition kernel
  2. TCP/IP (started and connected to the OpenEdition address space)
  3. DCEKERN

DCE configuration is performed through the program DCECONF with the appropriate parameters installed and environment variables set (see *Program Directory for OS/390* and *OS/390 OpenEdition DCE Administration Guide*, SC28-1584).

DCECONF is a TSO command that displays first a DCE LOGIN panel on which *Cell Admin ID* and *Cell Admin Password* must be provided.

After successful DCE LOGIN, DCECONF displays the main menu with five options:

1. Configure Server Machines to configure:
   a. The Security Server
   b. A Replica of the Security Server
   c. The Audit Server
   d. The Password Management Server
2. Deconfigure Server Machines to deconfigure:
   a. A Replica of the Security Server
   b. The Audit Server
   c. The Password Management Server
3. Configure DCE Client Machine
4. Deconfigure DCE Client Machine
5. Reconfigure Local DTS Entity
   a. As a Clerk
   b. As a Local server
   c. As a Global server

All the DCECONF functions are accessed through ISPF panels and are documented in *OS/390 OpenEdition DCE Configuring and Getting Started*, SC28-1583.

Each DCE configuration produces an output in the *dceconf.log* which is created in the home directory of the administrator who performed the configuration.

For DCE administration, OS/390 OpenEdition DCE provides *DCE Control Program* dcecp. dcecp is built on a portable command language called *Tcl.* It performs most DCE administration tasks using the component control programs. dcecp also provides *tasks objects* to perform a stream of complex DCE operations in a script (for example all the operations needed to add a host in a cell). For more detail, see *OS/390 OpenEdition DCE Administration Guide*, SC28-1584. In a multiple cell configuration, these scripts can be used to propagate a common cell configuration throughout the enterprise.

4.2.3 RS/6000 - AIX Environment

When installing DCE on a RS/6000 System with AIX version 4.1.3 (or higher) you need to install the following components on your “master” system:

dce.cds	DCE Cell Directory Services
dce.client	DCE Client Services DCE Client Tools DCE DFS Client Services
dce.compat	DCE SMIT
dce.doc	DCE Online Documentations
dce.ipfx	IPF/X for DCE Online Documentations
dce.pthreads	DCE Threads Library
dce.security	DCE Security Services
dce.tools	DCE Administration Tools

Use the `installp` command or the `smit install` functions to install this components. In addition you can install the following components, which may be used during enhanced cell definitions:

dce.tools	DCE Application Development Tools
dce.xdsxom	DCE X.500 API Library
dce.dfs_server	DCE DFS Base Server
dce.edfs	DCE DFS Enhanced Server
dce.dfsnfs	DCE NFS to DFS Authentication

Notice

During DCE installation and customization several files are used within the /var file system.

You will probably want to create a new AIX JFS file system in order to use DCE effectively. You should reserve about 30 megabytes for the initial cell definition. It should be mounted at /var/dce.

If the machine will be configured as DFS client, additional disk space is required for DFS cache files located at /var/dce/adm/fs/cache. If you do not want to create a separate file system, you should enlarge /var/dce. To make sure that /var/dce (and associated file systems) is sufficiently large, you should monitor its space usage.

To create a DCE cell in a simple way on an AIX system use as root the mkdce command. A sample command and output is shown in Figure 12 on page 45.

When prompted, enter the password to be assigned to the cell administrator. This will be required when performing other configuration tasks.

To get the cell alive, no additional steps are required on this machine. As you can see in the sample output in Figure 12 on page 45, all related client services have been created.

Another way to define the DCE cell is to use SMIT. The advantages are:

- All activities are logged in the file /smit.log.
- The command syntax need not be known when performing cell definitions in different ways - for example: having the Security Server and CDS Server on different machines.

For more details about using SMIT for defining a DCE cell, refer to *DCE V2.1 for AIX: Getting Started*, SC23-2797 or use the AIX DCE online documentation of *DCE for AIX Getting Started* and *DCE for AIX Administration Guide* while entering the following command:

```
>> /usr/lpp/dcedoc/bin/start_dcedoc
```

After system restart, you may either use the command line entry rc.dce all to start all DCE daemons, or use the SMIT interface.

To get DCE automatically started after a system reboot, an entry for /etc/dce/rc.dce should be inserted into the /etc/inittab. Use SMIT to add or delete this entry by calling the fastpath

```
>> smit mkdceitab
```

```
>> mkdce -n jans_cell.itsc.pok.ibm.com sec_srv cds_srv dts_local

Enter password to be assigned to initial DCE accounts:
Re-enter password to be assigned to initial DCE accounts:

Configuring RPC Endpoint Mapper (rpc)...
RPC Endpoint Mapper (rpc) configured successfully

Configuring Security Server (sec_srv)...
Password must be changed!
Configuring Security Client (sec_cl)...
Security Client (sec_cl) configured successfully

Security Server (sec_srv) configured successfully

Configuring Initial CDS Server (cds_srv)...
Configuring CDS Clerk (cds_cl)...
Password must be changed!
Waiting (up to 2 minutes) for cdsadv to find a CDS server.
Found a CDS server.

        Initializing the namespace ...
           Modifying acls on /.:
           Creating /.:/cell-profile
           Exporting cds-clerk and cds-server attributes
           Modifying acls on /.:/subsys/dce/sec
           Modifying acls on /.:/cell-profile
           Modifying acls on /.:/lan-profile
           Modifying acls on /.:/hosts
           Modifying acls on /.:/sec
           Modifying acls on principal ...
           Modifying acls on principal/krbtgt ...
           Modifying acls on principal/hosts/risc36 ...
           Modifying acls on group ...
           Modifying acls on group/subsys ...
           Modifying acls on group/subsys/dce ...
           Modifying acls on org ...
           Modifying acls on policy ...
           Modifying acls on /.:/sec/replist
           Modifying acls on /.:/risc36_ch

Initial CDS Server (cds_srv) configured successfully

CDS Clerk (cds_cl) configured successfully

Configuring Local DTS Server (dts_local)...
Local DTS Server (dts_local) configured successfully

Current state of DCE configuration:
cds_cl       COMPLETE   CDS Clerk
cds_srv      COMPLETE   Initial CDS Server
dts_local    COMPLETE   Local DTS Server
rpc          COMPLETE   RPC Endpoint Mapper
sec_cl       COMPLETE   Security Client
sec_srv      COMPLETE   Security Server (Master)
```

Figure 12. Cell Definition Using Mkdce Command

4.3 Further Cell Configuration

When talking about “further cell configuration” we will show the implementation of other systems into an existing cell, creating the required or additional services, or changing them.

Notice

Keep the following in mind when you configure a cell:

- Make sure that the machine’s clock is within five minutes of the cell’s master security server’s clock, otherwise authentication errors may occur, and configuration may fail.
- When configuring *DFS Fileset Database* machines and *Backup Database* machines, the clocks of these machines must be within 10 seconds of each other.
- When reconfiguring a particular component (or an entire machine), you must unconfigure (remove) the existing configuration before setting up the new one.

4.3.1 Client Configuration

Configuring clients requires *cell administration* as well as *system administration* authority. If the *cell administrator* is different from the *system administrator*, the configuration has to be done in two parts, where:

- The *admin* part requires *cell administrator* authority.
- The *local* part requires *system administrator* authority.

4.3.2 Client Administration

The *admin* portion of the client configuration can be performed by the cell administrator from any machine in the cell.

Easy administrative definition of the client is possible from:

- The AIX systems using the SMIT fastpath

  `smit mkdceclient`

  and then selecting the **admin only configuration** option.
- The OS/2 WARP system using the DCE configuration menu and selecting **Administrative** at the *specify configuration type* menu.

Entering the hostname of the DCE client machine and selecting the client types **sec_cl** and **cds_cl** (Registry and Namespace for OS/2 WARP) will update the namespace entries and the registry database.

4.3.3 AIX Clients

The DCE client software is part of the AIX system software for AIX Version 4.1, and requires no separate ordering.

To install the DCE client software on the client machine use the `installp` command or the SMIT fastpath command:

```
smit install_latest
```

Select the components **dce.client** and **dce.compat** (DCE Client Tools and DFS Client Services) and start the installation.

4.3.3.1 Local Client Configuration

After the *admin* portion is completed, start at the client machine as root SMIT with the fastpath:

```
smit mkdceclient
```

and select **local only configuration** option. You will be prompted to enter the name of the cell and the hostname (or IP address) of the master security server and names of the clients you want to configure. Possible clients are:

- RPC Endpoint Mapper (*rpc*)
- Security Client (*sec_cl*)
- CDS Client (*cds_cl*)
- DTS Client (*dts_cl*)
- DFS client (*dfs_cl*)

DTS client and DFS client are optional, but DTS client is recommended. If you want to also configure the DFS client, you have to enter information about the DFS cache. You can reserve data space for the cache in the memory of the system or you may define a certain amount of disk space for it (see also notice on page 44).

When the input is completed, press Enter to create the selected clients on the local machine.

4.3.3.2 Full Client Configuration

If you are both the *cell administrator* and the *root user* you perform a *full* client configuration from the client system, which incorporates both the *admin* and the *local* part of the configuration.

Starting the *full* client definition does not differ from *admin* or *local:* definition function except selecting the “full configuration” option. The input is the same as for the *local client configuration* but it will create the client services on the local machine as well as the namespace entries and the updates for the security registry database.

4.3.3.3 Client Configuration on Server Machines

When creating a Server function like Security server or CDS server on the local machine, the related client will also be created.

But if you have decided to have CDS Server, Security Server and DTS Server on different machines, you need to configure the missing clients. This will be handled in the same way as the *local* or *full* client configuration, except the clients for the installed servers must not be selected.

The DFS client, if wanted, should also be created on a DFS server machine.

4.3.4 OS/2 Clients

Client definitions on OS/2 WARP systems can be started using the DCE configuration functions. You can select, similar to the AIX DCE functions, *administrative, local*, and *full* client definitions.

For *local* and *full* you will see the same menus as for server definitions. The input is similar -- you should:

- Enter hostname and IP address of the “master” registry and name-server instead of defining the local system.
- Select component definition the *client - only* option.

For *admin* definition you will see a different menu that allows you only to enter the hostname and IP address of the machine to be defined.

Updates to namespace entries and to the registry database are handled the same as to client definitions in AIX.

The DFS client will be created automatically, as soon as you create any other client or server.

4.3.5 AS/400 Implementation Overview

IBM DCE Base Services/400 offers the DCE services that are designed to enable an OS/400 host to participate in DCE as a DCE client machine. The OS/400 host cannot act as a Security server, Cell Directory server, or as a Distributed Time server. The following daemons are provided:

- RPC daemon
- Security client
- CDS clerk
- CDS advertiser
- DTS clerk daemon

In order to run the IBM DCE Base Services/400 configuration program, access privileges have to be set up:

1. The Security Officer must enable the QDCEADM profile to allow a login as QDCEADM user, which is the local DCE administrative user id for AS/400.
2. The *cell administrator* has to be specified as a member of the subsys/dce security group.

There are two configuration-related commands on the OS/400 system; CFGDCECLNT to configure the system and DSPDCECFG to display the system configuration. They can be invoked by different ways, either by entering them as line commands or by using the main menu.

For *Configuring the Client*, in both cases a menu will be displayed to enter the required information such as cell administrator’s name and password, name of cell, CDS host name, Security servers host name, and so forth.

The *Display DCE Configuration* output menu will show the base cell information and the status about the configured components on the local host.

4.3.6 VM/ESA Implementation Overview

The implementation of DCE in VM/ESA OpenEdition supports only DCE client functions. It covers three main parts:

- Daemon Processes

  These processes provide DCE services to have the VM/ESA OE system running as DCE client. These processes run as individual POSIX processes within a dedicated virtual machine called *DCECORE* to initiate and control the daemon processes and to restart them if a daemon process fails. DCECTRL, an administration utility, allows the monitoring and controlling of the daemon processes from another virtual machine other than DCECORE.

  Components of these daemon processes are:

  - RPC daemon
  - Security client
  - CDS clerk
  - CDS advertiser,
    solicits and advertises the names and the status of all CDS servers in the network.
  - DCE Control Task,
    initiates, terminates and oversees the other daemons
  - Byte File System (BFS),
    a hierarchical file system used to support POSIX standards, and is also used by DCE.

- Runtime Support

  This is an API library to support DCE applications, services and utilities.

- Utility Programs

  These are used for developing DCE applications and administering DCE. These utilities run as POSIX programs within the applications.

  Important utilities are:

  - *EUVSUBFS EXEC*
    To build the static and variable file tree in the BFS.
  - *DCECONF EXEC*
    To configure and set up the OpenEdition DCE for VM/ESA.
  - *SKEWCALC EXEC*
    To check the time differences between VM and the target DTS server. The calculated skew value can be added into file `/opt/dcelocal/etc/time.skew`. Adding, or subtracting (if negative), this skew value is done to recalculate the DCE time, which must be within five minutes of the time provided of the DTS server.

4.3.7 OS/390 OE Client

The components and the requirements are the same as for the DCE server on OS/390 (see 4.2.2, “OS/390 OE Environment” on page 41). DCE client configuration is accessed through the *DCECONF* program with option 3 on the main menu panel. Cell name, security and CDS servers information must be provided to the program. DCE client configuration gives also an output in the `dceconf.log` file.

The DFS client function is not supported with the current version of DCE for OS/390 OE.

4.4 Server Configurations

The following section will provide a brief description of the DCE services and their components. Included is also some information to configure them.

Within Table 5 you can see which servers can be defined on which IBM platform and in which combination.

Table 5. DCE Server Combinations

	CDS	CDS	SEC	SEC	SEC	GDA	DTS	DFS
AIX	M	S	M	S	S	√	L/G	F/D
OS/2	S	M	-	M	-	-	L/G	-
OS/390	-	-	S	S	M	-	L/G	F

Where: M = master server
S = secondary server (replica)
L/G = local/global time server
F = file server
D = file location database

Server definitions are not possible on OS/400 systems and VM/ESA systems.

4.4.1 Directory Services

The DCE Directory Service provides directory service at the cell and global levels. The DCE Directory Service can store, retrieve and manage information about resourses (objects) such as computers, printers, users, files and applications. 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. The Directory Services has three main components:

- **The Cell Directory Service (CDS)**
  A CDS server stores and maintains object names within a cell to create and modify data. It also handles requests from CDS clerk for data look up.
- **The Global Directory Agent (GDA)**
  Is a “gateway” between local and global naming environments. It supports cell interoperability by allowing CDS to access a name in another cell using a reference that is stored in either the Global Directory Service (GDS) or the Domain Name System (DNS), a widely used global naming environment.

- **The Global Directory Service (GDS)**
  Supports the global naming environment between local and remote cells. GDS is an implementation of the X.500 standard.

  Note: GDS is not provided with the current IBM versions of DCE. However GDS can be used to locate other cells if it is provided by another product (such as AIX DCE 1.3).

4.4.1.1 Cell Directory Service (CDS)

CDS is the component that looks up and manages names within a cell. Client applications send requests through a *CDS clerk* process; if the data is not cached, the requests are passed to one or more servers to be handled.

The CDS server stores names (objects), which are unique within the cell, and their *attributes* in a database called a *clearinghouse*. The *clearinghouse* is tree structured and has directories that can contain further directories or leaf objects. Directories are the units that are used to distribute names throughout the cell namespace. CDS offers the ability to replicate CDS names (store copies of them on more than one clearinghouse).

An *attribute* is a piece of information related to an object, which can describe an object's class, network address, or other values. An object's name does not need to be changed if it is moved from one clearinghouse to another.

The following processes run on a CDS server machine:

- The CDS daemon, *cdsd*, is the CDS server process.
- The *cdsadv* which is used to receive and send server advertisements to find out what servers are available. On client machines, the *cdsadv* receives the server advertisements only.
- The *cdsclerk* daemon represents the CDS client process which is available in each DCE machine. The CDS clerk handles requests from client applications to a server and caches the result returned by the server.
- The *gdad* daemon is the GDA server, which sends lookup requests for foreign cells and returns the results to the CDS clerk.
- The DCE control program (*dcecp*) used for CDS management and maintenance, and the *cdscp* program used to control and display information about CDS clerks and servers.

  It is possible to start the *dcecp* and *cdecp* from a server as well as from a client machine.

The initial CDS server will be created during the initial cell definition on either an AIX system or an OS/2 WARP system. You may create additional CDS servers in your cell for replication purposes.

4.4.1.2 CDS Replica

Directory replication is the way to provide faster or more reliable access to the CDS namespace information. It is also the most reliable way to back up this information. Replication is defined on a per-directory basis. Each copy of a directory is a replica. CDS periodically ensures that the contents of all replicas of a directory remain constant.

One of these is the master replica; the others are read-only replicas.

- The *master replica* is the first instance of a specific directory in the namespace. The master replica is the only directory that can be changed while creating, modifying, or deleting information. After you make copies of a directory, you can designate different replicas as the master, if necessary, but only one replica of a directory can be the master at a time.
- A *read-only replica* is a copy of a directory available for looking up information only. CDS updates a read-only replica only when changes have been made to the master replica.

In an IBM environment you may create secondary CDS servers on AIX systems or OS/2 WARP systems:

- On AIX use the SMIT fastpath

  `smit mkdceserv`

  Select the *CDS server* and then *additional* options to enter the required definitions.

  Note: The same software components for AIX DCE should have been installed as mentioned in topic 4.2.3, “RS/6000 - AIX Environment” on page 43. This are also the prerequisites to install any other DCE server.
- On OS/2 WARP systems, definitions are similar to the initial definition (see 4.2.1, “OS/2 WARP Environment” on page 40); you use the *DCE configuration menus* called by the DCE icons.

When the secondary CDS server is created, only the *root* and the */.:/subsys/dce/sec* directories are replicated. In order for replication to happen, you must define every detail by explicitly creating the replicas, and defining the replica set with a master. You must also define the *skulking interval* that is used to copy a directory′s contents to all read-only replicas (for more details refer to the *DCE Administration Guide*).

4.4.1.3 Global Directory Agent (GDA)

To find names outside of the local cell, CDS clerks must have a way to locate directory servers in other cells. The GDA is an intermediary between CDS clerks in the local cell and CDS servers in other cells. The Global Directory Agent (GDA) enables intercell communications by serving as a lookup facility into the global naming environment.

A CDS clerk treats the GDA like any other name server passing it name lookup requests. However, the GDA provides the clerk with only one specific service. It looks up a cell name in the Global Directory Service (GDS) or Domain Name System (DNS) namespace and returns the results to the clerk. The clerk then uses those results to contact a CDS server in the foreign cell.

A GDA must exist inside any cell that wants to communicate with other cells. It can be on the same system as a CDS server or it can exist independently on another system. You can configure more than one GDA in a cell for increased availability and reliability. Like a CDS server, a GDA is a principal and must authenticate itself to clerks.

CDS finds a GDA by reading address information that is stored in the CDS_GDAPointers attribute associated with the cell root directory. Whenever a GDA process starts, it creates a new entry or updates an existing entry in the CDS_GDAPointers attribute. The entry contains the address of the host on which

the GDA is currently running. If multiple GDAs exist in a cell, they each create and maintain their own address information in the CDS_GDAPointers attribute.

When a CDS server receives a request for a name that is not in the local cell, the server examines the CDS_GDAPointers attribute of the cell root directory to find the location of one or more GDAs.

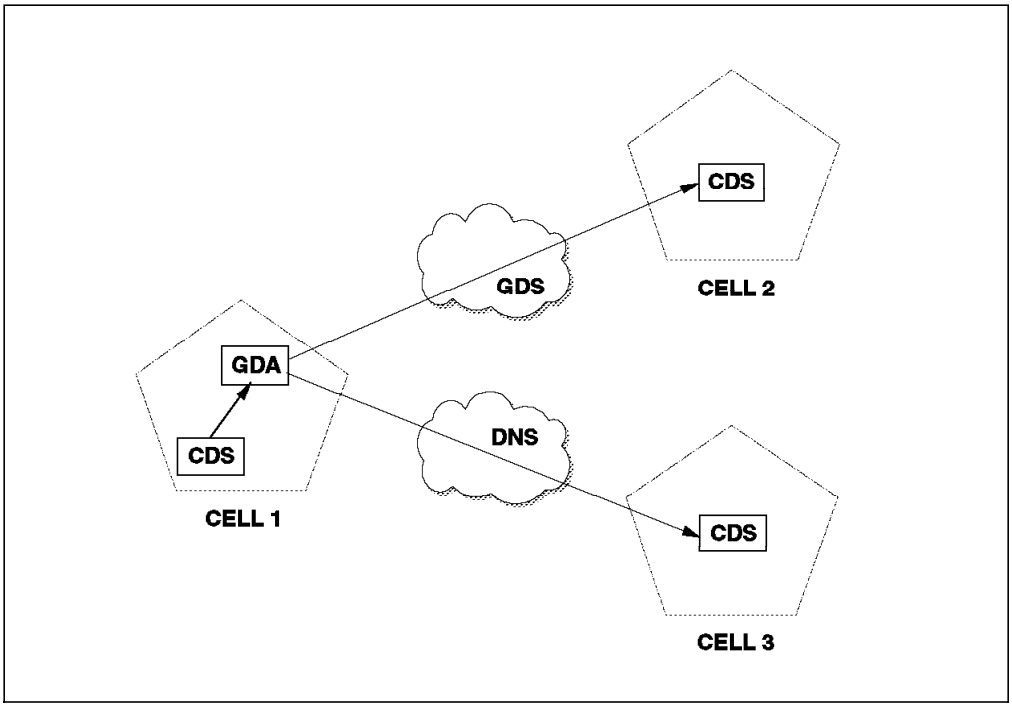


Figure 13. Interaction of CDS and GDA.
The GDA helps CDS to resolve names in other cells that are registered in GDS or DNS.

To enable authenticated communication between the local and foreign cells perform the following steps:

1. Use the SMIT fastpath on any node of each cell affected:

   `smit mkdceregister`

   to create a file that will be used as input for the Domain Name Server (DNS). These files should be copied to the system where the nameserver for the domain of the cell resides and must be added to the configurations of the domain.

```
;BEGIN DCE CELL /.../mini_cell.itsc.pok.ibm.com INFORMATION
;Initial CDS server
mini_cell.itsc.pok.ibm.com.   IN   MX   1  risc36.itsc.pok.ibm.com.
mini_cell.itsc.pok.ibm.com.   IN   A    9.12.0.36
mini_cell.itsc.pok.ibm.com.   IN   TXT  "1 8c121f7c-8d8f-11cf-870a-
1000a4f4c65 Master /.../mini_cell.itsc.pok.ibm.com/risc36_ch
8b378088-8d8f-11cf-870a-10005a4f4c65 risc36.itsc.pok.ib
;END DCE CELL /.../mini_cell.itsc.pok.ibm.com INFORMATION
```

Figure 14. Mkdceregister Sample Output. Default location is /etc/named.data.

2. To set up and start the *gdad* daemon use the SMIT fastpath

   `smit mkdcesrv`

   and select “GDA (Global Directory Agent) Sever” option.

3. Create in each affected cell principals for the foreign cells with the following command:

   `rgy_edit cell /.../foreign cell`

Note: With the current availabe IBM implementations of DCE, the Global Directory Agent can only be set up within an AIX environment.

4.4.1.4 Global Directory Service (GDS)

The DCE Global Directory Service (GDS) is a directory service implementation based on the international standard CCITT X.500/ISO 9594. GDS can serve two functions:

1. It can tie independent DCE cells from a worldwide directory service.
2. It can be used as an additional directory service to CDS for storing object names and attributes in a central place.

Like the Cell Directory Service, GDS is a distributed database which can be replicated. Each Directory System Agent (DSA), which is the server side of the GDS, stores a different part of the database. A DSA can cache copies (replicas or shadows) of the information from other DSAs to increase availability and performance. The information of the GDS can also be cached.

The client side of a GDS configuration is known as Directory User Agent.

Note: GDS is not provided with the current IBM versions of DCE. However GDS can be used to locate other cells if it is provided by another product (such as AIX DCE 1.3).

4.4.2 DCE Security Services

In an open environment where any number of systems can interoperate with each other, security becomes one of the most critical aspects in terms of authorization, authentication, secure communication and auditing. Security standards have been developed for security within networks. The DCE Security Service is one method used to meet these requirements.

The DCE security service ensures controlled access and secure communication with distributed systems. It enables clients and servers to provide their identities to each other. It also offers integrity and privacy of communications and supports controlled access to resources and applications. Therefore security is one of the main reasons to install DCE on systems within an open network.

The following Figure 15 on page 55 shows the main components of the DCE Security Server and how they work together.

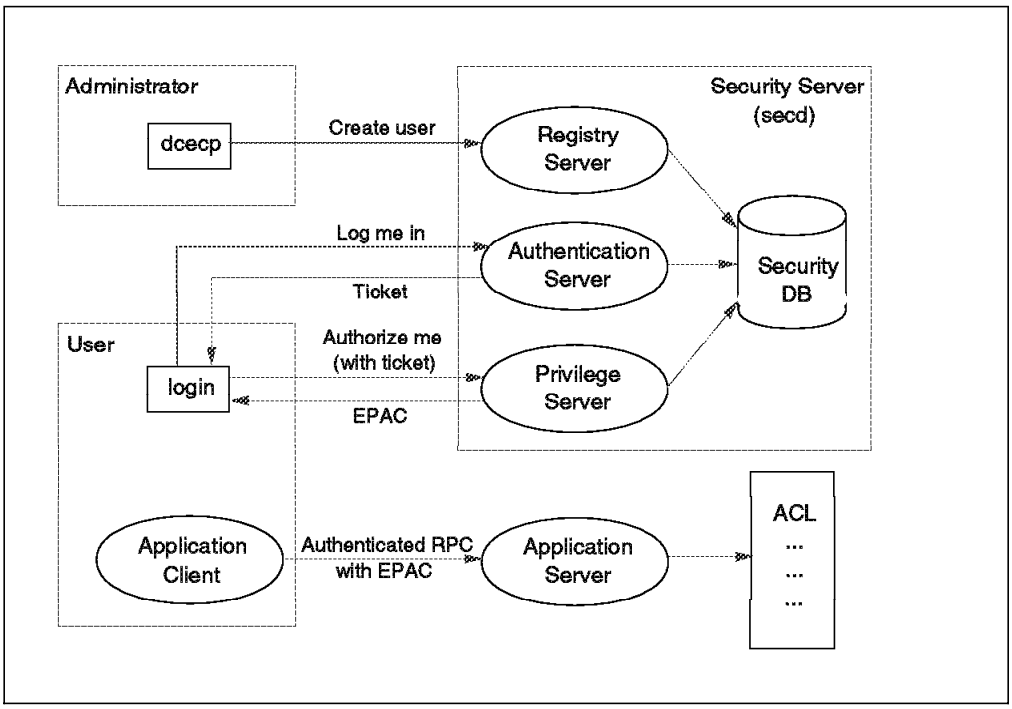


Figure 15. DCE - Security Services Functions

Here the main components of the DCE Security Services in more detail:

- *The DCE Registry Service* maintains a Registry Database.

  The registry service enables the administrator in a distributed network to define and maintain all security related information. The registry information objects are:

  principals The users of a system, human users as well as servers, machines and cells.

  groups Collections of principals identified by a group name.

  organizations Also collections of principals, but identified by an organization name.

  accounts Represents users related to principals. These objects contain accounting information and the password; they allow principals authenticated access to objects within the local cell and to objects in different cells (if defined).

  policies and properties
  Regulations about password length and format, also contains other authentication requirements.

  replist Used to manage replicas of the registry database.

  xattrschema Contains definitions of extended registry attributes (ERA). Can be defined as *schema entries* or *attribute types* which allows instances of attribute types to be added.

  The registry objects and attributes are stored into the registry database. The registry database is located in

  `/opt/dcelocal/var/security/rgy_dat`

  For load balancing, backup, and availability, it is recommended to create replicas of the security server and the registry database.

Only one replica in a cell, the *master replica*, accepts changes to the database. The master replica then propagates any update to the *slave replicas*. Updates are also saved in a log file on disk to ensure that no updates are lost in case of an unexpected restart of a server.

- *The DCE Authentication Service* component ensures that only a certified principal can login the system.

  At Login time, the security client uses the password the principal supplies to derive the principal's authentication key.

  A copy of of the principal's authentication key exists also in the registry database, prepared when creating the principal's account (or when the password has been changed). This key is used by the Authentication Service when decrypting to authenticate the principal. If the decryption succeeds, the keys are the same, the principal is therefore authenticated and login is successful.

  The DCE authentication service is based on *kerberos*, a secret-key encryption technology from Project Athena at the Massachusetts Institute of Technology.

- *The DCE Privilege Service* component ensures that those who are using the system have the necessary permissions to perform the operations they request.

  The Privilege Service certifies a principal's identity and group membership by providing a Privilege Attribute Certificate (PAC), which represents the network authorization of the client. PACs can be used with an Access Control List (ACL) to determine a principal's permission to access objects protected by a server.

- *The DCE Access Control List Facility* determines a principal's access to objects.

  Access to DCE objects is controlled by an authorization mechanism called an Access Control List (ACL). ACLs are associated with files, directories, Cell Directory Service (CDS) entries and registry objects. ACLs can also be used by applications to control access to their internal data objects. An object's ACL interacts with the protections provided by the object's UNIX mode bits. Each ACL consists of multiple entries to define the permissions of users (principals) or groups (organizations cannot be used for this purpose).

  DCE permissions, which are different depending on the type of the object, can be given to:

  - Owner, group and others
  - Specific individual principals in the local cell and in foreign cells
  - Specific individual groups in the local cell and in foreign cells
  - Any other principals in a specific foreign cell
  - Any principals in authenticated cells
  - Delegate users, servers or groups in local or foreign cells
  - Unauthenticated users

- *The DCE Login Facility* initializes a user's security context in DCE.

  The user interface for interactive principals is the `dce_login` / `DCELOGIN` command which communicates with the security server. Application clients and servers can use the `sec_login_validate_identify()` call to establish their

own login environment instead of using the identity of the principal that started them.

If the authentication attempt is successful, the Security Server returns the DCE credentials (privilege attribute data) within an initial ticket. During the principal's session these credentials authenticate the user who is using distributed services.

Notice

For AIX systems and OS/390 the administrator can activate options for a *single login* that allows to have the DCE authentication service included at the log in time for the operating system. No additional DCE log in is required (see also A.1, "Single Login" on page 89).

- *The DCE Audit Service* supports audit facilities for detection and recording of critical events in distributed applications.

  Audit plays a critical role in distributed systems. It is necessary to ensure that only authorized actions are performed. Audit, a key component of DCE, is provided by the DCE Audit Service.

  Note: On IBM operating system platforms, DCE Audit Service is currently only available with DCE for OS/390 OpenEdition.

  The basic components for the DCE Audit Service are:

  Application Programming Interfaces (API)
  to provide functions to detect and record critical events. An audit event is associated with a code point in the application server code. These APIs can also be used by tools that can analyze the audit records.

  Audit daemon (auditd)
  to perform the logging of the audit records based on the specified criteria. It maintains filters (who, what and when should be audited) and the central audit trail file.

  DCE control program (dcecp)
  the interface to the Audit daemon. It can be used to specify how the Audit daemon will filter the recording of audit events.

  The Audit daemon *auditd* does not need to run on all DCE hosts even if a client application uses the Audit service. The Audit daemon only needs to run on a host on which the log and filter files reside. Note that only the daemons on that host will have audit records recorded.

4.4.2.1 Planning the Security Server

Creating the security daemon *secd* together with the Registry Database as *master replica* is one of the first activities when creating a DCE cell (see also 4.2, "Initial Cell Configuration" on page 40). Also, when implementing a new system to a DCE cell, the security client daemon *sec_clentd* (AIX) will be created during the first step.

To increase the performance and the availability of your DCE installation, you may install an additional security server. The number of security servers in your DCE cell depends on your DCE environment: number of systems, network considerations and so forth. However, too many security servers in your cell

may also reduce your network performance when updating the replicas of the registry database.

While using the platform specific configuration tools, to configure a secondary (replica) security server you should select:

- **Secondary** when defining “SECURITY Server” from the *Configure DCE/DFS Server* menu called by `smit mkdcesrv` in AIX
- **Configure Replica Security server** from the *Server Configuration* menu called by `DCECONF` in MVS
- **Additional server and client** from the *Configure DCE Components on a Host* menu called by the DCE configuration icon in OS/2

With a secondary (or replica) security server a replica of the registry database (*slave replica*) will be established. The administrator does not need to configure anything, the replica database will be created and updated automatically.

The configuration of the *auditd* (audit daemon) can be performed on OS/390 OE systems using the *DCECONF* main menu while selecting option 3, (**Configure Audit server**).

The DCE control program `dcecp` can be used to display or modify the data in the registry database as well as for the Access Control Lists (ACLs).
In addition, the program `rgy_edit` can be used to access the registry database, while `acl_edit` is the program for accessing the Access Control Lists (ACL).

4.4.3 Distributed Time Service (DTS)

A distributed computing system has many advantages but also poses new problems. One of them is to keep the clocks on the different machines synchronized. Depending on the hardware, the system clock of each system drifts at a different rate. So without corrections, the system clocks throughout a network would become unsynchronized. The difference between any two clocks is known as *skew*.

The core services, especially the ticket granting service, rely on synchronized clocks. Uncorrected skew between system clocks within the network will influence the performance and availability of distributed applications.

DTS is a software-based service that provides precise, fault-tolerant clock synchronization for systems in Local Area Networks (LANs) and Wide Area Networks (WANs). The clock synchronization that is provided by DTS enables distributed computing applications to determine event sequencing, duration, and scheduling.

Most DCE nodes have a DTS clerk that adjusts the clock on its client system. The nodes that do not have DTS clerks have DTS servers. In addition to providing time values to clerks, servers also adjust the system clocks on their host systems. Servers are also able to obtain reference time values from sources of standardized time that are outside of the network or from external time provider such as radio, telephone or satellite.

In AIX you can use both *Network Time Protocol (NTP)* (an internet-recommended standard for system synchronization) and DTS in the same network, where NTP can act as a time provider for DTS or vice versa.

DTS offers a *Time-Provider interface (TPI)* that describes how a time provider process can pass Coordinated Universal Time (UTC) values to DTS server.

Figure 16 shows an example of the DTS components in a cell.

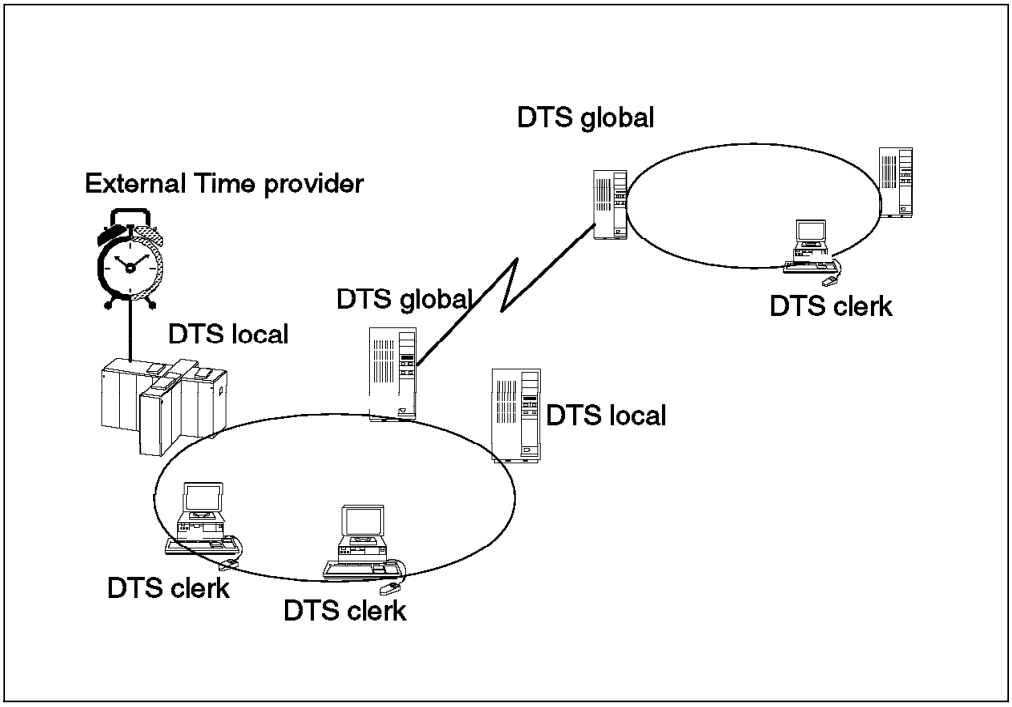


Figure 16. DTS Components. DTS Clerk and Server Roles

4.4.3.1 DTS Daemon

The DTS daemon *dtsd* can run as a client (clerk) or a server. DTS servers and clerks periodically synchronize the clocks in all network systems. The DTS daemon that is on each system performs this synchronization by requesting servers to send their clock and inaccuracy values. The daemon uses these values to compute a new system time. DTS servers and clerks have slightly different synchronization procedures.

Most systems in a DCE cell run the DTS clerk process. Clerks cannot have time providers and do not use their own system time to compute new synchronized times. The clerk uses only time values that it obtains from servers.

DTS servers provide many of the communications and synchronization functions for DTS. In addition to providing time values to clerks, servers also adjust the system clocks on their host system. Before attempting to synchronize with other systems, DTS servers always check if an external time provider is present on the servers system. If one is available, the server synchronizes only with the time provider. Otherwise it synchronizes with its peer servers while using its own system′s clock as one input value for the new time calculation. *Local servers* reside on the same LAN and are only available to servers and clerks within this LAN.

Global servers are also available throughout the network. They are necessary when:

- A network has multiple LANs or an extended LAN

- Systems in the cell have access through point-to-point links
- Clerks or local servers cannot access the required number of local servers determined by the minservers attribute of the dcecp dfs modify command.

The number of global servers is usually small, but global servers have important functions to enable DTS to synchronize all nodes in the network.

4.4.3.2 DTS Planning Guidelines

Planning your cell also is affected by the DTS implementation. Several rules generally apply depending on your network configuration and the number of nodes in the network.

Consider the following guidelines and questions when planning your DTS Implementation:

- Each cell should have at least three DTS servers in order to detect if one of them is faulty when they are queried for time. It is preferable to have more than three to provide redundancy. Additional servers increase the accuracy of time but also increase the activity on the network. The administrator must balance the level of accuracy with the amount of network activity.
- Locate DTS servers on the same nodes as the servers for the other DCE components whenever possible.
- Is your cell in a single LAN, an extended LAN, a WAN or a combination of them?
- How many servers will be required? Will global servers be required? Each LAN should have at least one server.
- Where will the servers be located? Locate the servers at the sites with the greatest number of different network connections.
- Will you use an external time provider to obtain Coordinated Universal Time (UTC)?
- If there are fewer than three time servers configured in the cell, the following commands should be used:

  ```
  dtscp set servers required n
  dcecp dts modify -minservers n
  ```

  (where *n* is the number of time servers).

4.5 Distributed File Service (DFS)

DCE Distributed File Service (DFS) is a distributed client/server application that presents DCE with a global view of a set of files and directories (a file system). DFS is considered as distributed because files can be stored on many different machines, independent from geographical locations, network, machine types or machine boundaries or operating system. They can be made available to clients regardless of where the files are stored. In the Distributed File System the user will see them as a single file system. This global view is called the DFS filespace.

The following Figure 17 on page 61 shows the data flow possible within a DFS filespace.

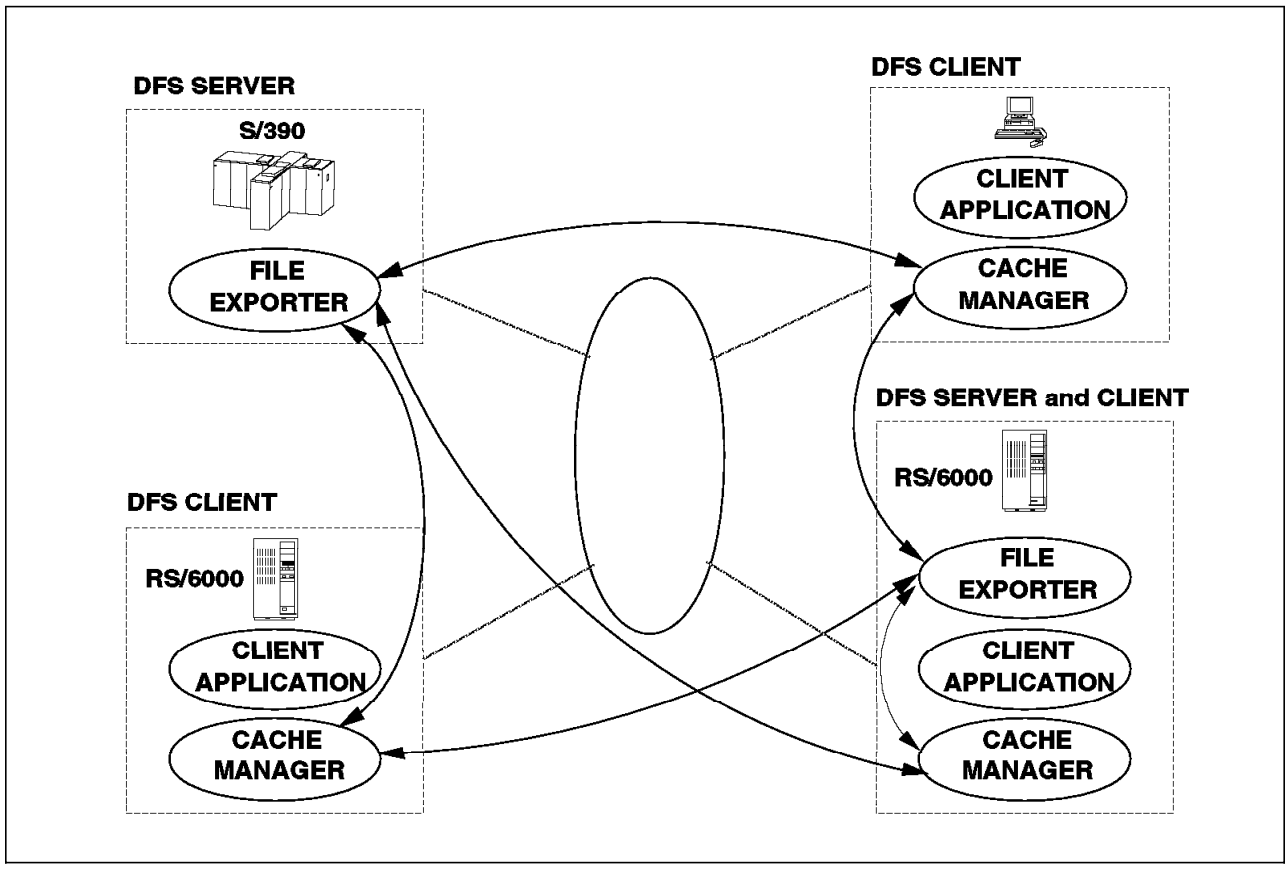


Figure 17. Data Flow in a DFS Filespace.

4.5.1 DFS Machines

4.5.1.1 DFS Server

This machine runs several processes to store or manipulate the data and to make the data available for clients. Other processes are required as interface to the DFS commands or are used for replication or backup. The Basic OverSeer (BOS) process runs on every DFS server machine. It monitors other server processes on the local machine and restarts failed processes.

These processes are available with the following server machines:

System Control machine Stores and distributes system configuration information, such as administrative lists, shared by all DFS server machines in the domain.

Binary Distribution machine Stores DFS binary files for process and command suites for distribution to all other server machines of its CPU/OS type in the cell. Note that the Binary Distribution Machine is not included in MVS DFS support. All the machines must be running the same version of the process for the system to perform correctly.

File Server machine Is used to store and export DCE Local File System (LFS) or non-local file system data for use in the global namespace.

Fileset Database machine Stores the Fileset Location Database which holds location information about filesets such as the name of the server where the fileset resides and the pathname used to reach the fileset.

Backup Database machine Stores the Backup Database. The backup database houses administrative information used in the DFS Backup System, such as dump schedules for backups. The information in the database is also used to restore the data in the event of a system failure.

4.5.1.2 DFS Client Machines

These machines are usually user workatations. The DFS client enables individuals to access DFS files and other general-purpose tools. A process called the *Cache Manager* runs on each client machine to request user data from the processes running on the File Server machine.

The Cache Manager stores a copy of the data in an area called the cache. The cache is a reserved area on a local disk or in memory of the client machine (see notice on page 44). Access to the locally stored file is much faster than access to the same data across the network. The data remains in the cache, so if the File Server machine becomes unavailable the client machine can continue to operate. The Cache Manager periodically sends the changed data back to the appropriate File Server machine to replace the data on the server. DFS advises all other Cache Managers with a copy of the changed file that their version is no longer valid.

4.5.1.3 DFS Configuration

DFS components can be installed and configured on the following IBM platforms:

OS/2 WARP - only client functions
AIX - client and server functions
OS/390 - only server functions.

Within the DCE cell, each system that has a need to work with data available in the DFS namespace has to be defined as a DFS client.

On OS/2 WARP (Beta) systems a DFS client will be configured automatically together with any other DCE configuration.

The DFS client definitions on AIX can be started together with other *full* or *local* client definitions (see also topic 4.3.3.1, “Local Client Configuration” on page 47 or topic 4.3.3.2, “Full Client Configuration” on page 47).

The definitions of DFS Server machines have some dependencies; some of which are:

- The number of machines which should distribute data
- Different types of hardware architectures and operating systems of the distributing machines
- Network configuration and performance

For AIX systems, the DFS software components can be installed on a local system without having the other DCE server components installed. The required components are the *dce.dfs_server* and the *dce.edfs* packages (see topic 4.2.3, “RS/6000 - AIX Environment” on page 43). The installation should be performed using the standard AIX installation tools *installp* or *smit install_latest*.

To configure the DFS server components you may use the SMIT fastpath

```
smit mkdcesrv
```

and select the DFS server machines you want to install on this system.

In OpenEdition DCE DFS with MVS/ESA as the operating system, you should define all DFS related information using the `rgy_edit`, `acl_edit`, `cdscp` and `rpccp` commands. A detailed description about the necessary commands and their sequence is shown in the manual *OE DCE DFS for MVS/ESA, Configuration and Release Notes*, SC24-5723.

Note: In OE DCE DFS on MVS/ESA, no DFS Fileset Location server can be created. This server must run on a non-MVS DCE DFS system.

Table 6 on page 64 gives an overview about the DFS machines and their roles. It also shows the name of the related processes and provides suggestions for implementation.

Table 6. DFS Machines and Their Roles

Machine Role	Purpose	Process	Suggestions
System Control machine	Distributes common configuration files for a domain.	bosserver upserver• upclient•	Use a Binary Distribution machine as the System Control machine for a domain.
Binary Distribution machine	Distributes system binary files for its CPU/OS type.	bosserver upserver• upclient•	Use the System Control machine for a domain as a Binary Distribution machine.
File Server machine	Exports and stores DCE LFS and non-LFS data.	bosserver ftserver fxd dfsbind repserver upclient• upclient•	A File Server machine must also have a server entry in the FLDB. In a large cell, dedicate one File Server machine to housing read-only replicas.
Fileset Database machine	Stores the Fileset Location Database (FLDB).	bosserver flserver upclient• upclient•	Configure three Fileset Database machines. Configure Fileset Database machines as Backup Database machines.
Backup Database machine	Stores the Backup Database.	bosserver bakserver upclient• upclient•	Optional, Configure three Backup Database machines. Configure Backup Database machines as Fileset Database machines.
DFS Client machine	Serves as a single-user or multiuser workstation; accesses files for application programs.	dfsd dfsbind	

• The Update Server that distributes common configuration files from a System Control machine.
• The Update Server that distributes binary files from a Binary Distribution machine.

4.5.2 DFS Local File System

DFS provides a high-performance, log based file system: the DCE Local File system (DFS LFS). DCE Local File Systems offer enhanced performance and reliability over traditional file systems by providing improved data storage and management. DCE LFS supports the use of *aggregates*.

A DCE LFS aggregate is physically equivalent to a standard UNIX disk partition, but it also contains metadata about the structure and location of information on the aggregate. In addition, a DCE LFS logs all operations that affect the metadata (file creation or modification). In case of abnormal system shutdown, the log can used to return the aggregate to a consistent state.

From a user's view, aggregates and filesets are the primary storage elements of the DCE Local File System. In a file system hierarchy, your files are contained in

directories, the directories are located on filesets and the filesets reside on aggregates (see Figure 18 on page 65).

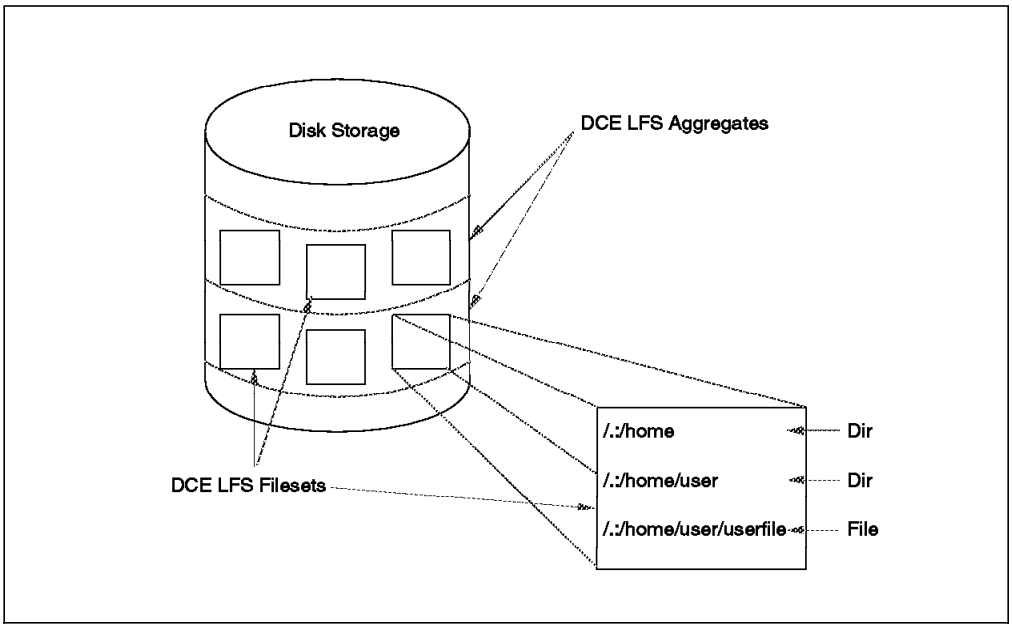


Figure 18. Aggregates, Filesets, Directories, Files.

The DCE LFS filesets can vary in size, but can never be larger than the containing aggregate (it would be impossible to be larger); multiple filesets could be stored on a single aggregate. Each fileset has a *fileset quota* that defines the maximum amount of data that can be stored in the fileset. The fileset quota can be modified by the system administrator to meet the requirements of the users.

To balance the load on the system across the available machines, the system administrator can easily move DCE LFS filesets from one aggregate to another or from one machine to another. As a DFS user, you never need to know the current location of your data. The Cache Manager of your client machines contacts the File Location Server to determine the location of your data.

4.5.2.1 DFS Replication

In DFS LFS, each fileset has a working (read/write) version. For performance, security, and availability purposes, DFS LFS filesets from the local machine can be replicated by creating read-only copies on multiple File Server machines. The data in a read-only version of a fileset cannot be modified, but it can be read and executed.

Different users can be given access to read-only copies of the fileset to prevent overloading one machine with requests for frequently used files. Replication also prevents the data being unavailable in case of failures at the system housing these files.

4.5.2.2 DFS Backup

DFS provides two methods of managing backups: the DFS Backup System and backup filesets.

With the DFS Backup System, you can copy filesets to tape and restore them in case the data is lost. Information about backups and tapes are stored in the backup database.

A *backup fileset* is a copy of a read/write DCE LFS fileset made at a specific time. Backup filesets do not reflect any changes made to data in the original fileset since the backup was created. It allows you to recapture a fairly recent version of your data without assistance from a system administrator. Backup filesets are read-only.

4.5.2.3 DFS and Non-LFS Data

Data from non-LFS file systems (such as Journaled File Systems (JFS) in AIX) can be used with DFS. You can export a non-LFS disk partition to the DCE namespace for use as an aggregate in DCE.

While an exported partition can be accessed in the namespace, it still holds only the single file system it contained at the time it was exported. Additionally, a non-LFS aggregate may not support features such as logged information about metadata, DCE ACLs, and fileset replication, which are available with DCE LFS.

4.5.2.4 NFS to DFS Authentication Gateway

Network File System (NFS) from Sun Microsystems is the most popular solution for sharing file systems. The NFS/DFS gateway is a product on the AIX platform that allows NFS client systems to access the DFS file space. The NFS client is typically run on a system that does not belong to a DFS domain.

The NFS/DFS Gateway can be installed on a DFS client system to export the DFS file system into NFS (acting as NFS server). The gateway provides a bridge between NFS and the authentication methods of DCE. It allows access to the DFS file space from an NFS client machine.

AIX DCE Version 2.1 also supports automated authentication from PC-NFS clients.

4.5.3 DFS Security

DCE security provides DFS with authentication of user identities and verification of user privileges and authorizations. When login to a DCE cell occurs, verified users receive authentication information in the form of a ticket. This ticket acts as proof to the DFS File Server of who the user is. The information in the ticket is used to determine privileges by comparing it with the authorizations granted in Access Control Lists (ACLs).

4.5.3.1 Access Control List in DFS

Using the DCE security's ACL mechanism (see page 56), DFS extends the standard UNIX permissions of directories and files (for DCE LFS filesets only). The DCE ACLs supplement the UNIX mode bits (read, write, execute) for DFS Local File Systems, they do not replace them. With DCE ACLs you can specify six different permissions for directories and four permissions for files, encoded as:

r	read
w	write
x	execute
c	control
i	insert
d	delete

The following Table 7 on page 67 shows which permissions are required for a specific task.

Table 7. Permissions Required for Specific Tasks. *Execute (x) permission is required on all parent directories.*	
Task	**Required Permissions**
Change a directory	**x** on the directory
List directory contents	**r** on the directory
List data on files and directories	**r** and **x** on the parent directory
Create a file or directory	**w,x,** and **i** on the parent directory
Delete a file or directory	**w,x,** and **d** on the parent directory
Rename a file or directory	**w,x,** and **d** on the current directory **w,x,** and **i** on the new parent directory
Read a file	**r** on the file
Write a file	**w** on the file
List ACLs on a file or directory	(x on parent directories)
Change ACLs on a file or directory	**c** on the file or directory

With DCE ACLs the sets of users to which permissions apply (user, group, others) in the standard UNIX environment have been expanded so that you have the possibility to define access authorization for many different users and groups (see page 56).

4.5.3.2 Administration Lists

DFS supports enhanced administration and security both by making DCE ACLs available for objects in the DFS LFS filesets and by using administrative lists with DFS server processes. You can create ACLs for users or groups of users to extend the same permissions or privileges to multiple users simultaniously. Because each server machine has his own administrative list, a fine granularity of control with respect to server process administration is possible.

4.6 Multiple Cell Definitions

The definition of a cell is given in 1.2.1, “DCE Cells” on page 3.

A multiple cell design requires the cell administrator to enable communication, directory and security services between the cells.

4.6.1 Network

On top of the physical links, a transport level service must be available, such as User Data Protocol (UDP), Transmission Control Protocol (TCP), or ISO TP0-TP4 transport protocols.

The systems must also have network protocols in common, such as Internet Protocol (IP) or Open Systems Interconnection (OSI) protocol.

4.6.2 Directory Services

If a user from cell A wants to access a server from cell B, the directories from cell A and cell B must be linked. This is done through the global directory service (GDS) and the global directory agent (GDA). GDS is a distributed database that registers DCE cell information. See Figure 19.

GDS conforms to the X.500 directory service standard.

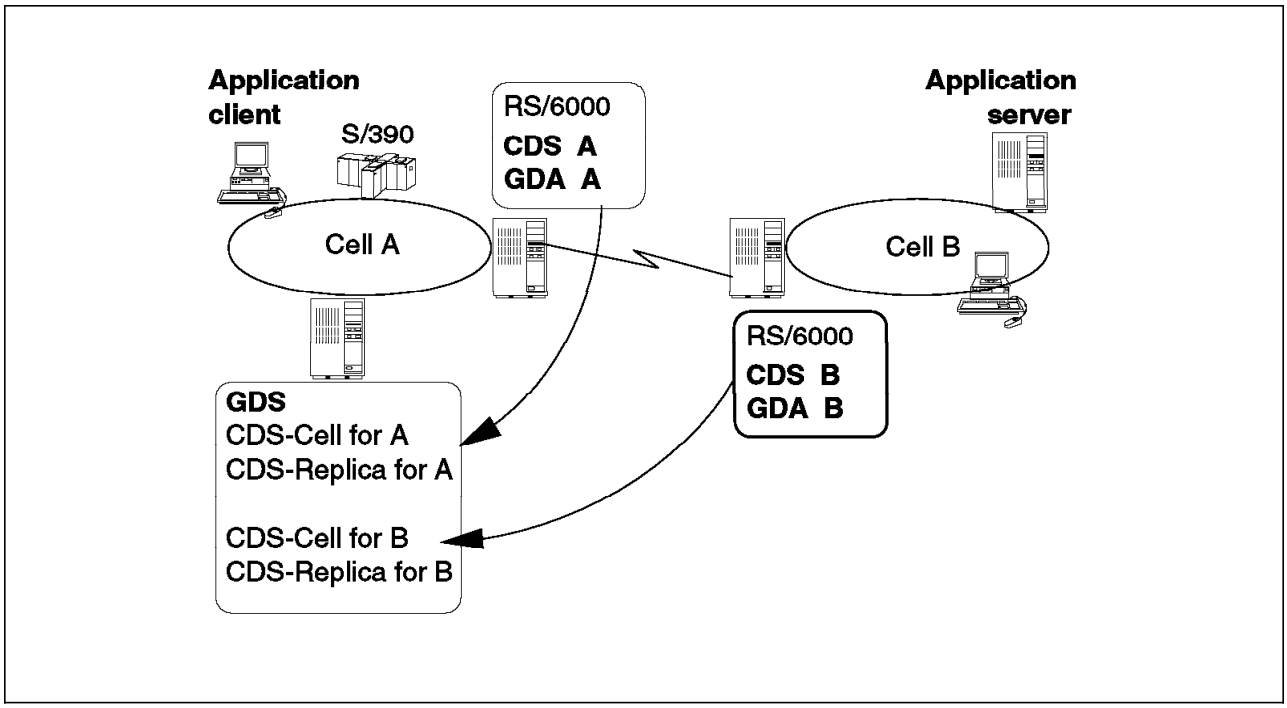


Figure 19. Directory Definition in Multiple Cells Configuration. Cell A and Cell B Entries in GDS.

GDA is a function that helps the CDS clerk of cell A find the location of servers that are not in cell A. The daemon that runs as a machine process to provide the GDA function is called the gda daemon (gdad). See Figure 20 on page 69.

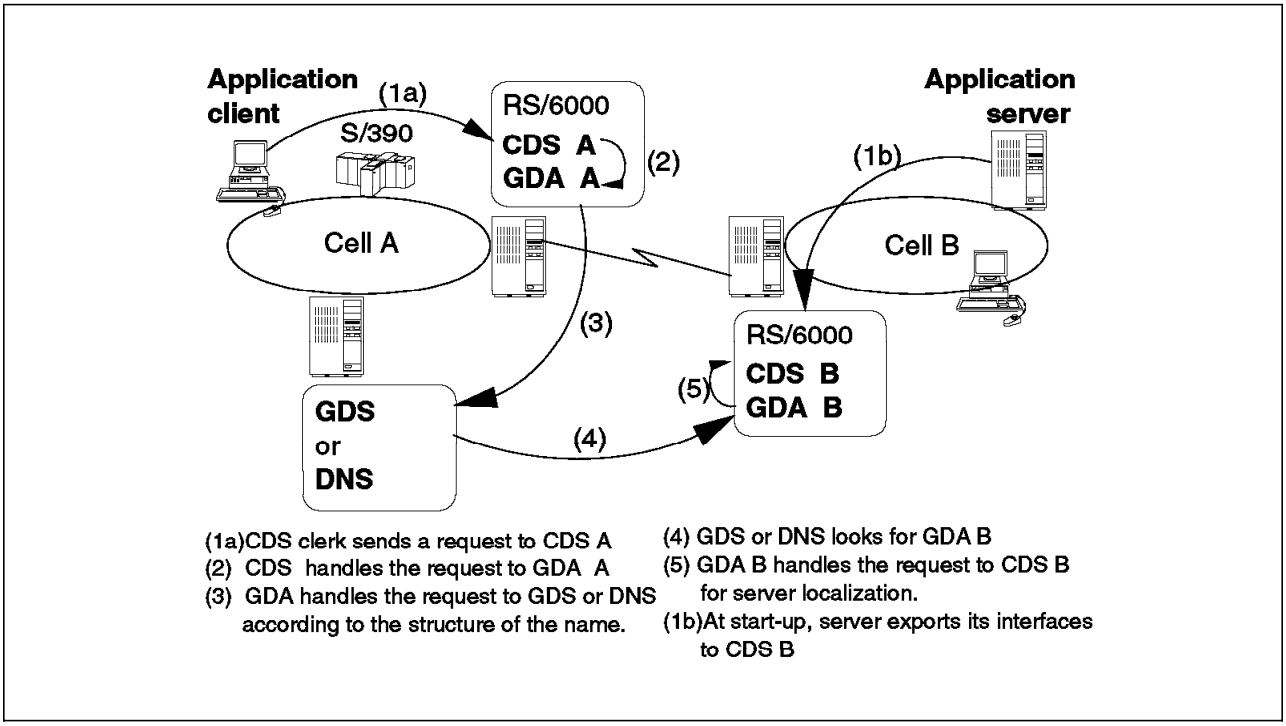


Figure 20. Cross Cell Naming Resolution

The cell administrator may prefer to register cell information in the Domain Name System (DNS), which is a function of TCP/IP. DNS supports DCE directory services but is not a DCE component. From the structure of the name, GDA recognizes if the request is to be sent to the GDS or the DNS.

4.6.3 Security

In a multiple cell environment, DCE security can be implemented across the cells.

If security authentication has to be performed when a client from cell A links to a server from cell B, trust relationships have to be established between the cells. The `rgy_edit cell` command creates two cross-cell authentication accounts, one in each cell. One password is shared by these two accounts.

The command can be performed on either of the two cells. You can also use the `dcecp registry connect` command. Information about the foreign cell (name, authentication account and password) has to be passed as parameters to this command. Please refer to administration guides for parameters and syntax:

- *DCE for AIX Administration Guide, softcopy*
- *DCE for OS/2 Administrator's Command Reference*, S96F-8505
- *OS/390 OpenEdition DCE Command Reference*, SC28-1585

See Figure 21 on page 70.

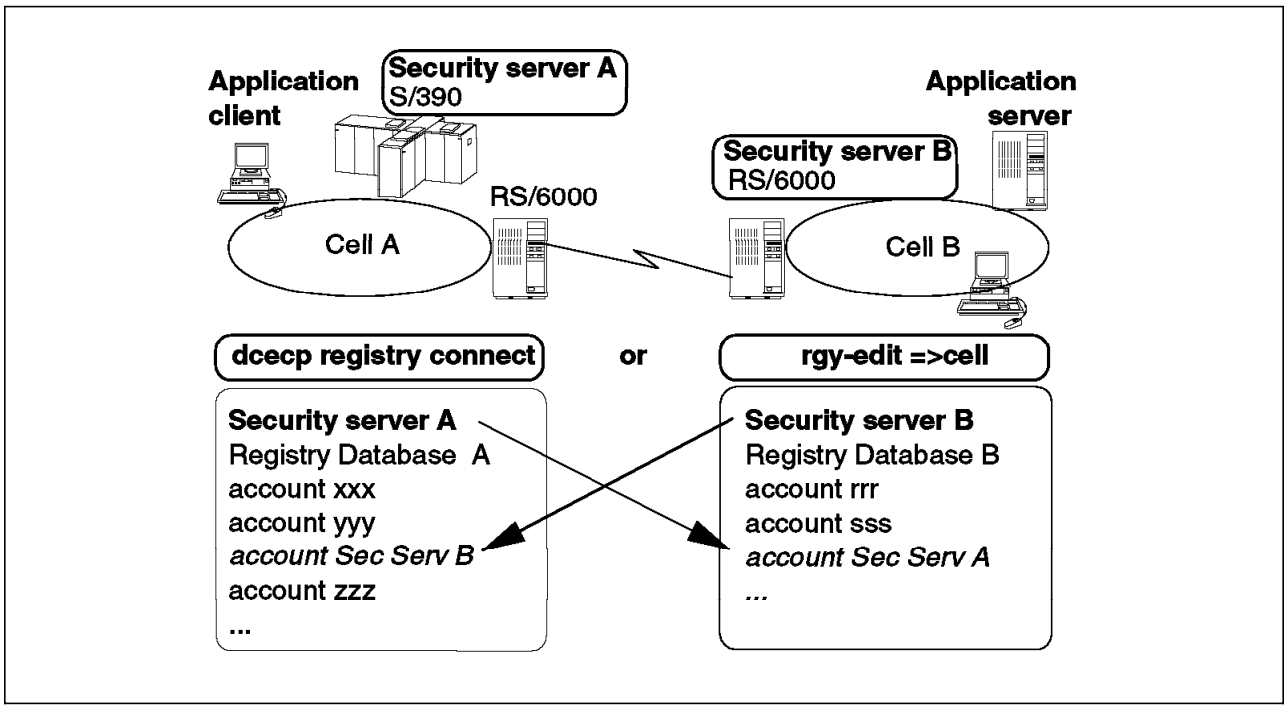


Figure 21. Security Definition in Multiple Cells Configuration. Cross Cell Trust Relationships Creation from OS/390 or from AIX.

This registration has to be done for each pair of DCE cells which will interoperate.

4.6.4 DFS Cross-Cell Definitions

Allowing “foreign users,” that is, users from a different cell, access to the DFS server, is done by setting ACLs properly.

- For AIX, in acl_edit an entry of type *any_other* has to be created in the local cell with at least *rx* permission (read and execute).
- For MVS, the fully qualified DCE name of each foreign principal has to be mapped to an MVS user ID. Each principal is entered in the “Identity Mapping Input File.” The administrator of the local (MVS) cell must have sufficient authority to add a *foreign_user* entry in the ACL of the */.:/sec/principal* directory of the foreign cell. Please refer to *OpenEdition DCE DFS for MVS/ESA Configuration and Release Notes*, SC24-5723-00 for more details.

Chapter 5. Application Implications

We review in this chapter some characteristics of *DCE application development*, what tools can be used to help in this process and then to manage the applications, and finally where DCE is used in IBM applications.

5.1 DCE Application Server Considerations

An application server must have several characteristics because it operates in a distributed environment. We can list here:

1. The ability to inform the members of the cell (clients and servers) of its location. The location of the server may vary for technical reasons. The server must give a means to the clients to find its new location. Changes must occur without service disruption for the client.
2. The ability to deal with network or system crashes, that is, error handling and recovery procedures. In a distributed environment, the client is not always informed that the server is down.
3. The ability to keep running and connected to the network, being ready to handle requests at any time.

Item 1 is directly handled by The application itself: it must export its interface and possibly binding handles to the CDS so the client will not only know *where* to access the application but also *how* to use the server.

Items 2 and 3 have a direct impact on the design of the cell. Several backup infrastructure strategies may be considered to insure application server nondisruptive operation:

- Cloning of the application server
  - Separate servers in the cell, all declared in the CDS
  - Coupled servers in the cell through a mechanism, such as HACMP, where a machine can take over the load of the server if it crashes

  Separate servers will apply in any cell design (single cell or multiple cell), while HACMP applies only to machines on a campus location.
- Duplication of network paths to the application server

These considerations show that the application server characteristics in a distributed environment will usually lead to a redesign of the application.

Application Support servers on MVS for CICS or IMS are the exception. A secure and robust environment such as MVS manages all the reliability aspects of the application servers without replication. Performance aspects are handled by the configuration and the tuning of the Application Support servers.
The usage of application support servers is the case where “legacy” applications (IMS and CICS) remain unchanged while access to them is opened to a new set of clients in a distributed environment. No major redesign of the legacy application is needed in this case.

5.2 DCE Application Development Process Overview

Once the choice of the appropriate client/server model has been done, (that is, how the process is distributed between the client and the server), the developer has to go through several steps that we will describe briefly:

- Interface definition
- Directory services utilization
- Security services utilization

The characteristics related to Application Support servers development is outlined.

5.2.1 Client/Server Interface Definition

Interface definition is a specific DCE application development step.

5.2.1.1 General Case

The DCE application development process contains a step not found in usual application development. This step consists of defining the client/server interface of the application. The definition is written using DCE Interface Definition Language (IDL). Then it is compiled. IDL is a declarative language with a syntax similar to the C language. It consists of a set of prototypes for the remote procedure call. The interface definition is then compiled with the DCE IDL compiler. The output of the compilation is:

- A pair of object files, the “stubs”
- A header file

There is *server stub* code that must be linked with the compiled server application code and the DCE library, and a *client stub* that must be linked with the compiled client application code and the DCE library. The purpose of the stubs is to provide to the application with the functions needed for remote execution. The header file, which follows the same process as the stubs, contains the data structure of the application and the interface specification identifier.

Once the interface has been defined, it can be implemented; that is coding of the application, initialization routines, coding of directory and security services, and so forth, is performed.

5.2.1.2 DCE Application Support for MVS/ESA

When the server runs on MVS/ESA or OS/390, the definition of the interface is also done through an Interface Definition Language with some extensions.

Although the process is the same (stubs and header are generated by the IDL), the interface is defined to allow DCE clients to access IMS or CICS transactions.

The client side is unchanged from the general case (client program is usually written using C language, then compiled and linked with client stub), while on the server side stubs are separate load modules and are dynamically loaded (see Figure 22 on page 73).

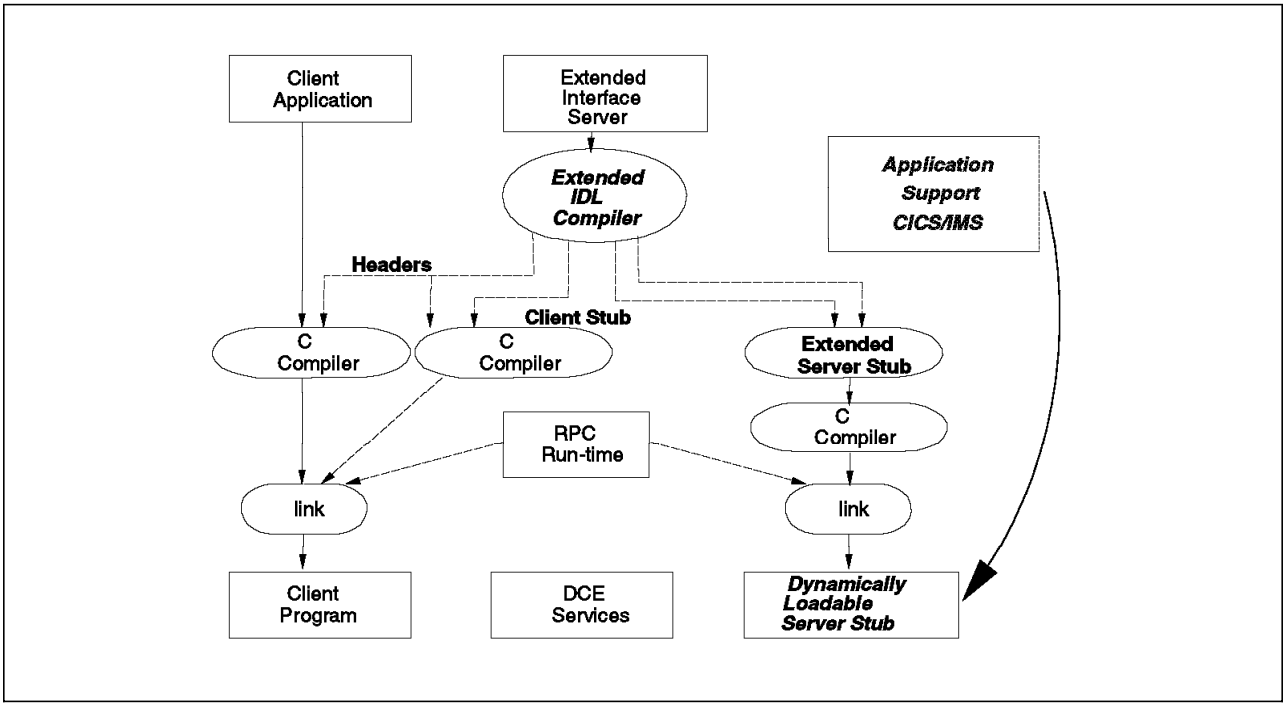


Figure 22. IDL Process Using Application Support for MVS/ESA. Servers stubs interface to CICS or IMS transactions.

The application support program brings several advantages to the development process:

- Existing programming skills are utilized in the development of new client/server applications, since IDL allows a programmer to use COBOL parameters for server functions.
- CICS and IMS programs are written in COBOL; the client code may interface to already developed transactions or new ones.
- Application support converts between COBOL and C data types.

The existing usages of IMS and CICS transactions are not affected by the new interface to DCE application support.

In terms of utilization,
application support servers allow any DCE client application to access OS/390 applications.

The DCE client has the choice of the network through which he is connected to the OS/390:

- An SNA network; in this case, the *Anynet* feature is required on VTAM to convert SNA protocol to TCP/IP. Also an *anynet* gateway is required on the local network of the client station or on the client workstation itself. In OS/2, this function is fullfilled by *Multi-Protocol Transport Service (MPTS).*
- A TCP/IP network; in this case the connection to the OS/390 system can be:
  - Through a 3172 controller connected to a channel
  - Through an OSA adapter on the system itself

On OS/390, VTAM, TCP/IP and DCE software are required.

For the case of DCE Application Support for MVS/ESA, the server application may be an existing COBOL program or a new COBOL program used to process IMS or CICS transactions. See Figure 23 on page 74.

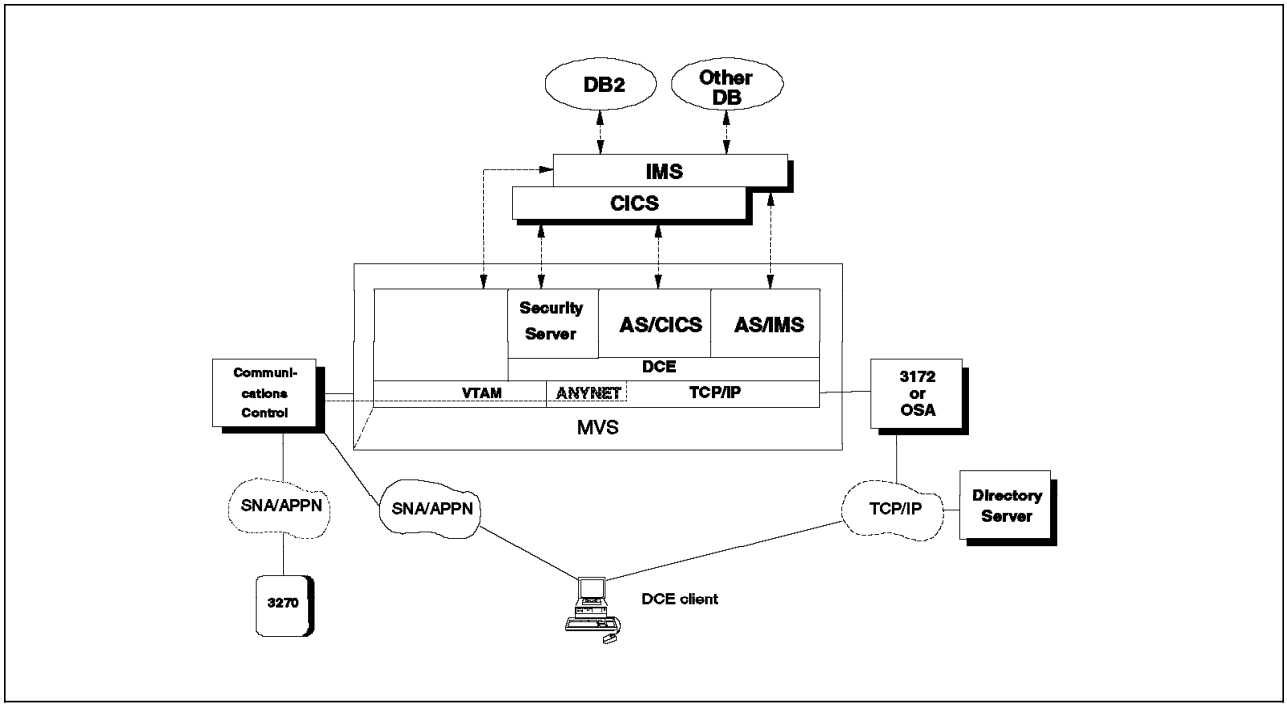


Figure 23. Application Support Servers on MVS for IMS and CICS

A detailed description of application support servers can be found in *OpenEdition DCE Application Support for MVS/ESA, Configuration and Administration Guide*, SC09-1659 and in *OpenEdition DCE Application Support for MVS/ESA, Programming Guide*, SC09-1530.

5.2.2 Directory Services Utilization

The CDS database will store names and location information allowing clients to find the servers.

5.2.2.1 Naming

The developer uses a naming structure which reflects the organization of its applications and separates the different functional groups. This is done by configuring the CDS namespace entries for the application.
An application should always export its interfaces to CDS when it starts and remove them when it stops.

As with any DCE application server, IBM Application Support servers has its namespace entries in the CDS configured using cdscp commands. The Application Support server is defined in the CDS using the administration program ASUADMIN (ISPF panel guided program). For details see *MVS/ESA OpenEdition DCE Application Support Configuration and Administration Guide*, SC09-1659-00.

5.2.2.2 Binding Handles

Binding handles contain the information the *calling code* needs to link to the *called procedure*. A binding handle contains:

- The network protocol information
- The network address of the server
- The listening port of the server (endpoint)
- The interface id of the server

The binding handle is also exported in the CDS database.

Application Support server works in this standard DCE manner. There is a restriction: Application Support Server does not support one type of binding called *automatic binding*.

5.2.3 Security Services Utilization

To operate at a good level of security, the application must use security services to prevent intrusions on the systems and the network. Basic security will rely on *authentication*. Clients and servers will be registered in the Security Registry.

To secure the access to the resources, ACL routines will be coded to provide more advanced security.

Application Support servers are registered in the registry database. Application Support users must be registered in the DCE registry database and in the RACF MVS database as they access MVS resources. The interoperation between RACF and DCE (see 2.6.2.2, “RACF/DCE Interoperation” on page 18) will allow this mapping.

5.3 DCE and Object Technology

Object technology allows programnmers to develop applications by assembling elementary blocks of computing. These blocks of code and data are highly reliable because they have been used and tested many times. Object technology insures a high level of reusability of the code and this leads to easy to maintain applications.

Distributed object technology adds to the components described above the capability to be distributed in a network and executed on different systems. In a distributed environment, the interfaces to access the objects, and the services they can use, are the result of a specification work led by the *Object Management Group (OMG)* in the *Common Object Request Broker Architecture (CORBA)*.

IBM implements the CORBA standard through the *System Object Model (SOM)* and *Distributed SOM (DSOM)*. SOM is supported by IBM′s operating systems platforms (AIX, OS/2 Warp, OS/390, OS/400), development tools, and languages (Visualage, Smalltalk, COBOL, C/C++ and so forth).

CORBA has already defined an object-oriented naming specification. This namespace specification resembles the DCE Cell Directory Service, which is an object name resolution process that can be based on a CDS search mechanism. The CDS infrastructure can support the object naming definitions, but does not in the currently available version of OMG naming.

Another CORBA interoperability specification between objects uses the DCE RPC.

Time and security services specifications will be defined by the OMG in 1996.

DCE can offer the infrastructure for communication and services to the object-oriented technology. Both DCE and object technology are part of IBM's strategy in application development.

5.4 DCE Development Tools

Basic DCE development is probably not the easiest and the fastest way to get DCE applications implemented. We review in this topic some of the tools (and list the providing vendors) that increase productivity in the development process.

- Entera from Open Environment Corporation (OEC)
- Connection/DCE by Open Horizon, Inc.
- Visual-DCE; by Gradient Technologies, Inc.

5.4.1 Entera from Open Environment Corporation (OEC)

Entera is the follow-on product of Encompass from OEC. It provides APIs to DCE simpler than standard DCE APIs. The three-tiered approach of the tool lets developers build and change applications efficiently and rapidly by separating them into three components:

- The user interface (client)
- Application logic (server)
- Data access technology

The developer station is supported by several operating systems such as:

- AIX
- OS/2
- Windows
- NT
- HP-UX

and can use GUI development tools that have an interface with Entera. Among these are:

- Enfin on OS/2 TCP
- PowerBuilder on DOS/Win
- Visual Age on OS/2
- Visual Basic on DOS/Win,
- Visual C++ on DOS/Win and NT

Recently announced are some new additions to the company's Entera product lines which include *Entera Client for Open Edition MVS*, *Entera/TransAccess for MVS* and an *Entera port to MVS OpenEdition*.
Entera Client for Open Edition MVS, automates the development of client communications code that links GUI development tools such as PowerBuilder, C, Visual C++, and Perl to DCE servers running under MVS OpenEdition. This new Entera client allows customers to build desktop applications to run on Windows, Windows NT, Macintosh, and OS/2, even if those systems are not running DCE.

With *Entera/TransAccess for MVS*, distributed clients can access MVS servers and MVS servers can in turn be clients to other applications and data sources outside of the MVS environment. Unlike gateway products which allow read-only access to the database, Entera for MVS can invoke CICS and IMS applications that perform read/write access, and utilize data integrity rules as part of the distributed environment.

5.4.2 Connection/DCE by Open Horizon, Inc

Connection DCE is the first database connectivity product that offers full integration with DCE services. Connection/DCE is a modular database connectivity solution that allows organizations to implement 2-tier, 3-tier, gateways, transactional and non-transactional client/server applications through the common database interfaces.

Connection/DCE offers developers the benefits of platform and DBMS independence, eliminates the need for users to learn new languages, and incorporates DCE security and directory services.
Connection/DCE supports Informix, Oracle, Sybase, DB2 and other data base management systems.

5.4.3 Visual-DCE by Gradient Technologies, Inc

Visual-DCE provides an object-based view of DCE that minimizes the complexities of writing DCE applications for Windows environments.
It utilizes Gradient's PC-DCE to access DCE services from the Windows platform.
Visual-DCE is layered on Microsoft's Visual Basic programming system and includes Visual-DCE custom controls that simplify creating RPC interfaces, RPC bindings, and DCE Login Contexts. Developers can add these controls to their programs in the same way they add other Visual Basic controls. They can modify a control's property values via the Properties dialog box, or by making changes directly in their program code. Visual-DCE automatically translates remote procedure arguments between C and Visual Basic and encapsulates the DCE layer through a dynamic link library (DLL) to isolate your application from the DCE RPC implementation.
No direct calls to any DCE API are required.
Visual-DCE also translates DCE exceptions into error codes that can be captured by Visual Basic and handled accordingly.

5.5 DCE Administration Tools

We also review some administration tools to help in the management of a cell.

- Distributed Access Control Manager by Dazel Corporation
- DCE Cell Manager by HaL Software Systems
- Doxa Distribution Tool Kit (DDTK) by Doxa Informatique

5.5.1 Distributed Access Control Manager by Dazel Corporation

DACM (Distributed Access Control Manager) is a replicated DCE-based access control service. DACM's centralized control over privileges complements the security services provided with DCE and offers an alternative to developing separate ACL Managers for each DCE application service deployed.

5.5.2 DCE Cell Manager by HaL Software Systems

The DCE Cell Manager is an integrated set of graphical user interface tools meant to help DCE administrators organize, monitor, and control access to DCE services.
Cell administrators can accomplish their tasks from a single machine rather than moving from machine to machine or remotely logging into system after system. The DCE Cell Manager eliminates the command structure of OSF DCE and replaces it with a consistent graphical representation.

The DCE Cell Manager toolset consists of:

- Namespace Manager
- Host Configuration manager
- User and Group Security Manager
- Support and services to ease the move to distributed computing

5.5.3 Doxa Distribution Tool Kit (DDTK) by Doxa Informatique

Doxa Distribution ToolKit is a collection of middleware tools using OSF DCE threads, RPC and CDS (DTS and security optional). *DDTK* features high performance (typical applications can handle hundreds of requests per second), load balancing between servers, server redundancy, one way and two way queuing mechanisms, support for orderly shutdown of applications, optimization for low bandwidth networks. DDTK Application Builder allows developers to build distributed C or Cobol applications by graphically defining service interfaces and server queues. Application screen interfaces are built with a GUI tool. Application servers may function with any database (relational or object-oriented).
Tuning DDTK application client/server links is also performed through a graphical interface. DDTK offers helpdesk operators a powerful graphical tool to display real-time information on the communication channel between the user application and the server, to help locate problems.

Note

This is only a selection of the numerous tools available in the DCE environment. We give a general description of what they claim to do. We do not intend to compare these tools between them or with others that are not mentioned in this book.
OSF provides a catalog of the products which currently use and/or implement DCE. This catalog is on-line on the internet at
`http://www.osf.org/dce`

5.6 IBM Applications Using DCE

The applications IBM has developed during the past years are implementing the Open Blueprint Architecture. In this architecture middleware functions such as directory, security and time are direct implementations of the DCE architecture (see 1.2, “DCE Architecture” on page 3). We review in this topic how IBM applications use these DCE services.

5.6.1 IBM Software Servers

IBM has announced a family of modular application servers, known as IBM Software Servers. These servers enable customers to rapidly implement client/server applications, and extend application capabilities to distributed environments. Most of these “Software Servers” use DCE.

5.6.1.1 IBM Database Server

In a distributed database environment, DB2 for AIX has its own directories to locate:

- Databases
- Nodes
- Database connection services

As an alternative, DCE directory services can be used to replace these directories.

Note: DB2 on MVS (DB2 V4R2) uses DCE security services when it is accessed in a client/server mode in a distributed database environment. DRDA functions allow the use of DCE tickets for authenticating database clients and are available over a TCP/IP network (please refer to IBM announcement letters for details and general availability date).

5.6.1.2 IBM Transaction Server

This server offers two different transaction monitors:

- IBM CICS for AIX
- IBM Encina for AIX (Encina is a family of products developed by Transarc)

The DCE services that CICS for AIX uses are:

- Remote Procedure Call (RPC)
- Cell Directory Services (CDS)
- Security Service
- Threads

The Cell Directory and Security Services of DCE, although recommended for multi-server environments, are optional, as CICS for AIX can be supplied with server host names and endpoints and also offers simple security services.

The Encina Monitor is IBM′s Transaction Server system for customers that have adopted DCE for their distributed computing infrastructure. Encina extends and enhances the DCE by simplifying application development and by providing a robust execution and administrative environment for deploying large scale client/server systems.

The client/server functions of Encina are tied closely to DCE. Application security is implemented using DCE access control lists; servers are located by clients using the DCE cell and/or global directory services; client-to-server communication is through the DCE RPC or Encina Transactional RPC.
By building upon DCE security, the Encina Monitor provides a very strong security implementation that provides authentication, authorization, and encryption.

5.6.1.3 IBM SystemView Server

Several features of SystemView server for AIX require a DCE infrastructure for their communications:

- SNA Manager for AIX, which is a network manager
- Extended Systems Administration, which is a system management tool
- Printing Systems Manager based on the Palladium distributed print system (see 5.6.4, “Printing System Manager”)

Please refer to the *SystemView Server for AIX* documentation for details.

5.6.1.4 Directory and Security Servers for AIX

Directory and Security Server is a packaging solution that includes all the DCE components necessary to build a DCE cell, including DCE tools.
Directory and Security Servers for AIX includes the following components that can each be installed on the same or different computers:

- DCE Directory Services
- DCE Security Services
- DCE Base Services
- DCE Tools, for example, IDL compiler, symbols and message strings utility, diagnostic tools

5.6.2 IBM MQ Series

In the message queuing process, when an application has a message to send, it opens a queue which is handled by a queue manager. With IBM MQ Series for AIX, queue location and names can be held in DCE cell directory server. This is a pluggable name service in the MQ framework. In this way, MQ framework provides interface to the naming service for OEM components.

5.6.3 Distributed Security Manager

DSM for AIX requires a full DCE infrastructure and manages security over the cells (see also 2.6.2.3, “Security Administration” on page 18).

5.6.4 Printing System Manager

IBM Printing Systems Manager (PSM) for AIX provides centralized management of distributed printing in networks with many printers. PSM uses DCE security, time, RPC, and directory services as defined in the Open Blueprint.
PSM uses DCE for security services to authorize command execution and printer access and provides great flexibility in limiting them to exactly what is required for each individual user.

PSM is a feature of SystemView Server for AIX.

5.6.5 IBM AIX LAN Distributed Platform/6000

The LANDP family of products, including LANDP/6000, has evolved from IBM FBSS (Financial Branch System Services) products, and provide excellent support for branch office environments in banking, finance, and other industries. These environments typically require support for industry oriented devices, peer to peer connectivity within the branch office, and interoperability with enterprise systems based on CICS IMS, DB2, MQSeries and non-IBM software running on a variety of platforms.
IBM AIX LANDP/6000 provides interoperability between the two distributed application environments of LANDP and DCE. It allows LANDP/2 and

LANDP/DOS applications to benefit from DCE technology. Additionally, applications using DCE, running on platforms where DCE is available, can benefit from all currently available LANDP services (LANDP-provided or user-written).

The implementation of the Open Blueprint Architecture is a permanent task in the development of applications by IBM. This list is intended to grow at the pace of new products announcements.

It is the growth of the implementation of DCE components in IBM and vendors software, as well as the ability to provide development and management tools over DCE, which will insure that DCE will increase its position in the distributed computing market.

Chapter 6. DCE on the Internet

This chapter shows several ways in which DCE may be used when access to the Internet is to be involved. The chapter is not meant to be an exposition of all possible application interfaces, nor is it meant to explain the most efficient means of Internet access; it is meant to simply show some of the possibilities when DCE applications are to be available and Internet access is to be provided.

6.1 Using a Server As a DCE Cell Gateway

This topic is meant to show how DCE applications can be made available to customers or potential customers from the Internet. One basic assumption here is that the full DCE security and application access facilities are not necessary for the data on customer transactions. A second assumption is that the users may be somewhat casual; they may have encountered the availability of the application through “surfing” activities, so the applications should be simply and easily accessible. Since potential customers may not wait for long transmissions, we cannot take their time to download full DCE client code on their system.

The gateway machine from the DCE cell to the Internet should not act as a cell server and the enterprise Internet interface. The enterprise Internet access may have a high load from users obtaining advertising material or general company information. This type of load should not be allowed to impact cell operations. One possible strategy is that the enterprise Internet machine pass DCE application users to a machine in the DCE cell, who acts as a client for the required application. The second possibility is that the enterprise Internet machine is in the cell, but limits DCE activity (no DCE server activity nor application server functions) to serving as a client for the Internet user applications. This configuration would probably be the one employed when CGI-BIN applications that require DCE services are accessible through form submissions or URLs available on the Internet.

As shown in Figure 24 on page 84, the Internet user in this case does not become a member of the DCE cell. The customer from the Internet accesses an application that provides information or an interactive service (for example, the ability to enter an order). The customer is not concerned that the application makes use of DCE and need not even be aware of it, since the DCE usage is all between the Internet “gateway” node and the rest of the DCE cell. The gateway machine performs the application and sends the results to the Internet user.

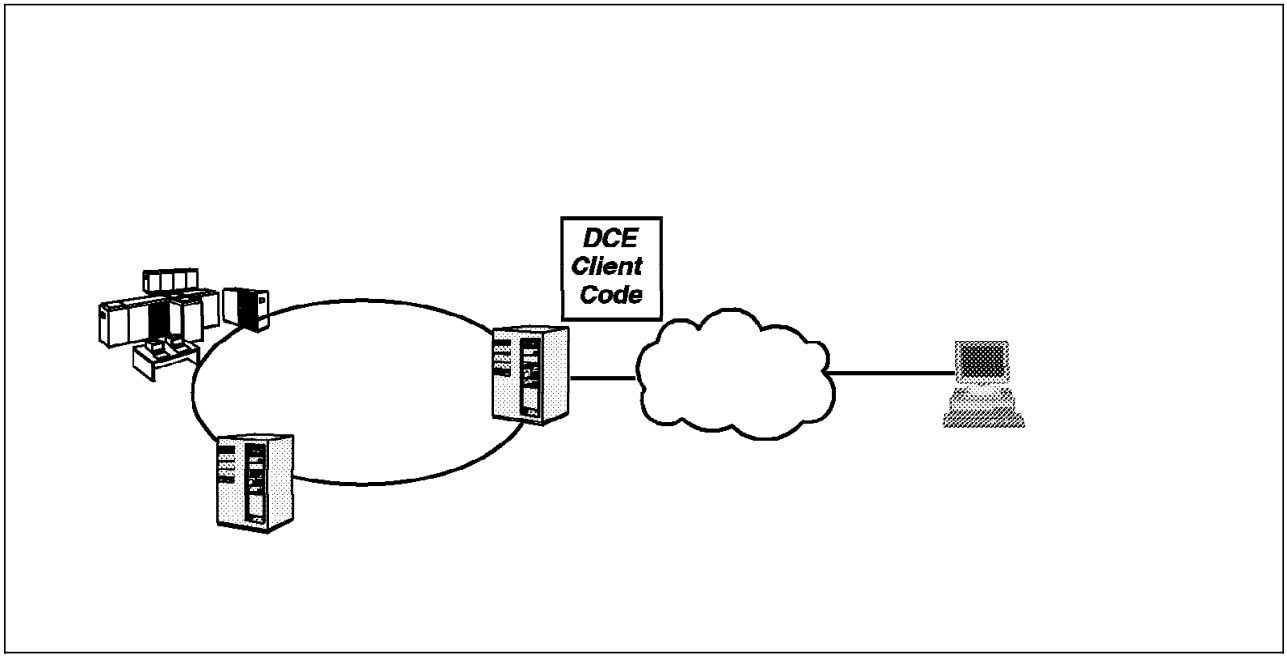


Figure 24. Gateway Access from an Internet Client

Any DCE client code necessary for the application resides in the gateway node, so no resources for that code are required in the application user's machine.

6.1.1 Security Considerations - DCE Cell Gateway Solution

Obviously, DCE security functions cannot be used for the data flow from the cell gateway to the user; neither the user nor his machine ever became a member of the cell. Some external means for encryption and password protection must be found with this solution. From a security standpoint, this solution should only be considered if reliable non-DCE security programs exist or the nature of the data is such that it need not be secure.

6.1.2 Other Considerations - DCE Cell Gateway Solution

With any Internet application, thought should be given to a heavy load that could be generated by browsers or surfers. Sites that were expecting a few users per day have experienced thousands; either due to being placed on a "hot" list of sites, being publicized in the news media, or including a word in its description that resulted in frequent hits by web-search programs. To reduce the cell load of such users, the enterprise DCE gateway should not be the Internet gateway. The Internet gateway should be a machine with the capability to direct DCE traffic to the DCE gateway, with the result that only users of the DCE applications would be executing on the DCE gateway. Those users surfing, or accesssing such data as advertising material would be satisfied by the HTTP server running on the Internet gateway.

6.2 Providing Customer Access As a DCE Node

This approach is to provide a higher degree of customer usage of DCE facilities than the approach presented in the preceding section. The application user in this case does need DCE facilities, so the time to load some DCE code is justified. The user then becomes a member of the DCE cell, rather than simply requiring an interface to an application that resides in the cell. As shown in

Figure 25 on page 85, the DCE client code is downloaded to the customer's machine, so enough storage space must be available for it and enough machine resources for execution of the client code must be available. Using this approach, organizations should make an effort to have pre-configured the client code; otherwise resources must be available for configuration execution. It should be recognized that these resources are used only while the application is to be executed. The storage and machine resources will be freed at the end of the usage.

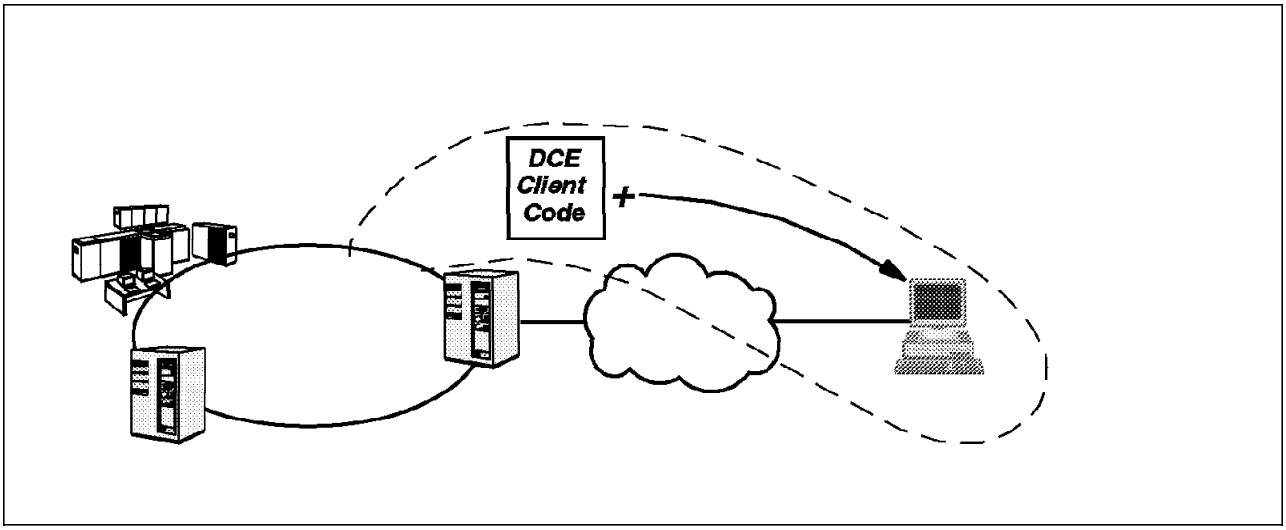


Figure 25. Providing DCE Code to an Internet Client

An example of this kind of usage would be the provision of a library of DCE applications to Internet users. The applications might reside on any DCE node, so Cell Directory Service (CDS) client code is used to locate and contact a server for one or more of the applications.

6.2.1 Security Considerations - DCE Client Download Solution

This method is no more secure than the preceding one. The data flow from the cell could be viewed by anyone at a link location in the network, so passwords or secure data could be accessed. Security could be more easily provided than the first solution by allowing the user to obtain a password through contacting the enterprise through a different medium than the transmission line (for example, through an 800 dial telephone number). Without the second-medium usage, this method should only be used for "internal net" testing or for transactions that have no security requirements.

In this case a second security consideration exists, namely that the client code and configuration data have been made available to the network. It is possible that this user could use cell membership at a later time to attempt unauthorized usage. It is also possible that a different user made a copy of the code, perhaps as it passed through an internet node on the way to the intended user. In either of the situations, the unauthorized usage or attempted usage would be made more difficult if the client privileges were immediately revoked at the end of the expected transactions. Revoking should also occur on any planned or unplanned ending of client communications; a user should not be able to maintain cell privileges by merely turning power off on his workstation. It is also implied that the user machine identification and password will be set differently each time the client code and data is transmitted.

6.2.2 Other Considerations - DCE Client Download Solution

The considerations of load imposed from Internet connection noted in 6.1.2, “Other Considerations - DCE Cell Gateway Solution” on page 84 apply to this solution. Note also that revoking membership will cause more server activity; each (planned or unplanned) user termination should cause priviledges to be revoked, which means updates to the security server and to the cell directory server will occur. Finally, you should recognize that machine clocks will not be reliably set, nor can proper time zone identification be assumed, so DTS should not be used in the cell unless extensive clock synchronization activity is planned.

6.3 Using the Internet for Connectivity to DCE Nodes

The Internet user in this case requires full DCE cell membership as in the previous case, but the difference is that this user regularly and frequently joins the cell through Internet connections. This user may be expected to have DCE code available. There may even be DCE applications available for other cell members from this location.

As shown in Figure 26, the Internet node in this case is expected to provide permanent machine and storage resources for the client facilities.

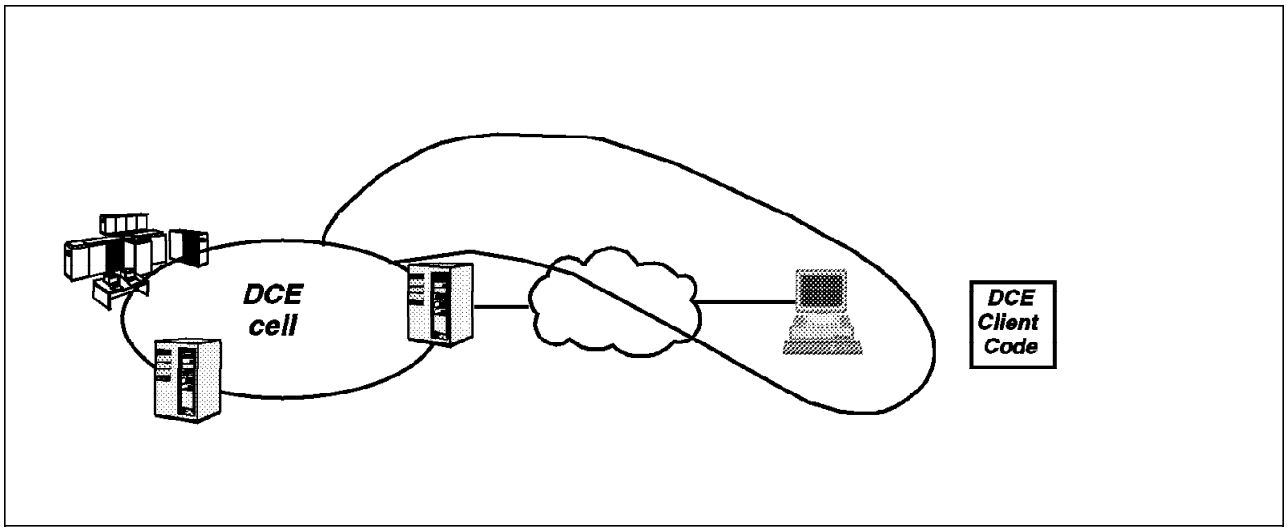


Figure 26. DCE Cell Member with Internet Access

Only in the case of maintenance upgrades would the code be transmitted across the network.

6.3.1 Security Considerations - Pre-Loaded Client Solution

This method can safely use DCE security features, provided passwords are not sent unencrypted across the network. (Note that this cannot happen in the DCE login process, so the consideration here is in providing usable passwords to users). The passwords could be obtained using the method of the previous solution (through a second medium). Another possibility for this solution is to provide pre-loaded code and passwords in the workstation as it is delivered.

6.3.2 Other Considerations - Pre-Loaded Client Solution

There is a major difference from a workstation connected by other means and one connected through the Internet: most Internet providers do not allocate a fixed IP address to their users, so the client configuration information regarding address may be required to be updated prior to joining the cell. This change must be performed in the cell servers as well as the client, which implies a “pre-login” procedure to inform the code both in the cell servers and in the workstation that an address change has occurred.

Appendix A. Features Supported in DCE Implementations

This appendix outlines some specific features found in DCE implementations on different platforms.

A.1 Single Login

The end user must usually login to several environments to complete his job:

- To the station on which he is working
- To the operating system of his station or of his server
- To his applications (login to DCE if they are DCE applications)

Each of these operations requires a user ID and a password, a situation that may not be comfortable for the end user. It also requires additional effort for the system and DCE administrator to maintain all this user ID's and passwords.

Single login, as it is implemented in AIX and OS/390, allows the possibility of having only one login operation for the operating system and DCE applications.

A.1.1 Single Login AIX-DCE

In DCE for AIX Version 2.1, the base operation security services have been integrated with the DCE Security Services. With this release of DCE we have the possibility to have a single-system image rather than separate images for DCE and AIX. AIX commands `login` and `su` as well as remote login commands (`rlogin`, `telnet`, `rsh`) allow the user to authenticate with the DCE registry. DCE users can also change their password using the AIX `passwd` command.

To activate the "single login" facilities, the system administrator should do the following:

- Verify that the module `/usr/lib/security/DCE` is installed on your system.
- Edit the file `/etc/security/login.cfg` and add the following lines:

```
DCE:
   program = /usr/lib/security/DCE
```

- Start the *dceuinixd* daemon. To get the daemon started after system reboot you can add this daemon to the `/etc/inittab` file.
- Edit the file `/etc/security/user` to define the authentication methods for the users; as in the following example:

```
auth1 = SYSTEM
SYSTEM = "DCE or (DCE[UAVAIL] AND compat)"
```

Notes:

- Existing AIX users should have the same user ID assigned when defining them to DCE.
- You may add new users (accounts) using SMIT DCE administrative functions `smit dcesecadmin`. Creating a principal and an account allows the new user to log in as a DCE user as well as an AIX user. The new user will not have an entry in the `/etc/passwd` file.

A.1.2 Single Sign-on OS/390-DCE

This function is one of the results of the interoperability between RACF and DCE. The principle is that DCE principal's name and password are available to OS/390 when it needs to log a user in to DCE. This information comes from a new RACF DCE segment which is associated with an OS/390 TSO user ID. With this information, an OS/390 user who has been authenticated in OS/390 by RACF can run a DCE program without reauthentication to DCE. See curved arrows in Figure 27 on page 91.

Practically, three tasks must be completed to enable the single sign-on:

1. The RACF administrator must defined a *DCE segment* for the OS/390 user in his RACF parameters.
2. In this DCE segment, a flag must be set properly that is: `AUTOLOGIN=YES` (default setting is NO).
3. The OS/390 user must store his *DCE password* in the RACF registry using the `storepw` command.

These operations are described in *OS/390 OpenEdition DCE Command Reference*, SC28-1585 and *OS/390 OpenEdition DCE Administration Guide*, SC28-1584.

Interoperability between RACF and DCE also allows DCE servers running in OS/390 to use OS/390 user IDs on behalf of their DCE clients. If the server is authorized, it can perform TSO login on behalf of the TSO ID associated withthe connecting DCE user.

Administration tools are provided to cross-link information between the DCE registry and the RACF database.

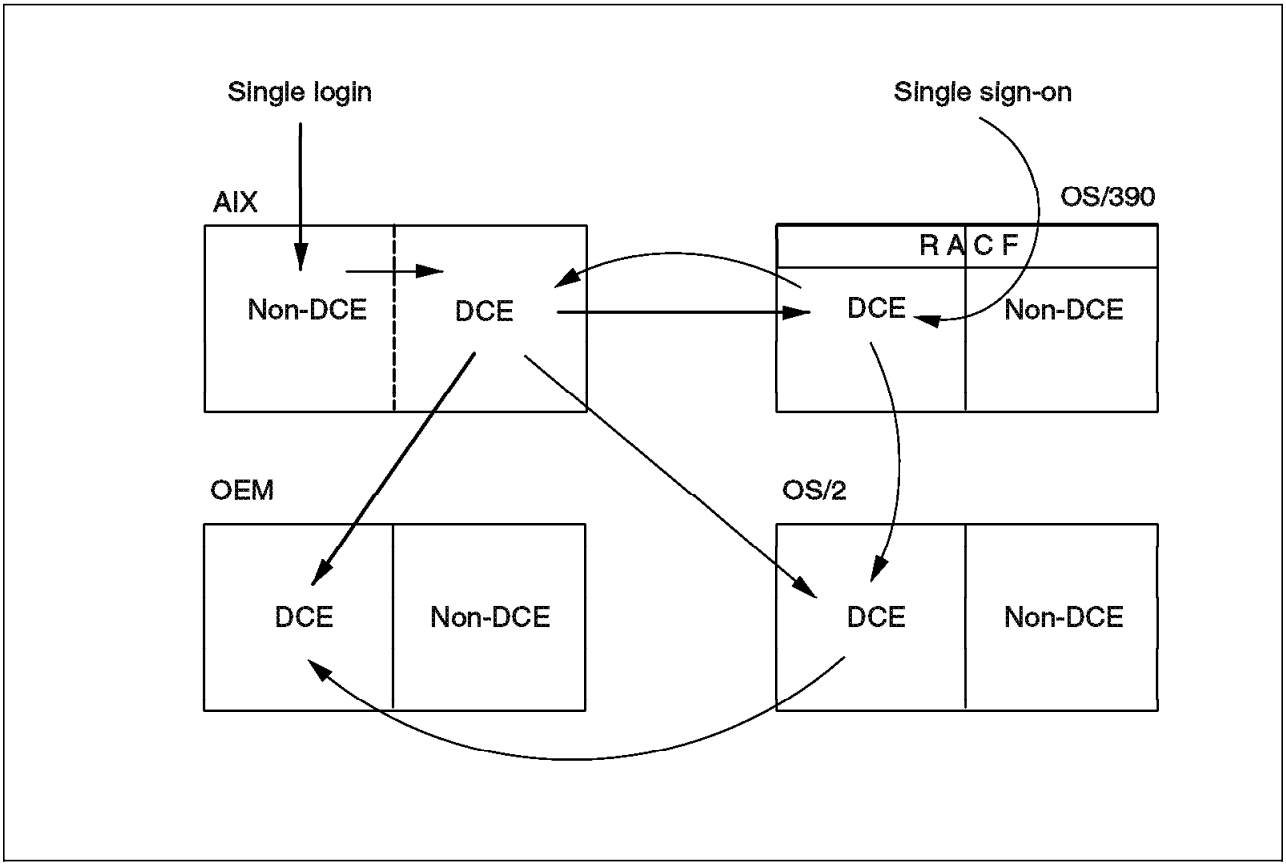


Figure 27. Single Login on AIX and Single Sign-On on OS/390.

A.2 HACMP and DCE

IBM AIX High Availability Cluster Multi-Processing (HACMP) represents an important product for RS/6000 system used to minimize down time by quickly restoring services in case a system, a component, or an application fails.

HACMP guarantees no single points of failure and operates in three modes:

- One machine (node 2) is sitting idle and watching the other one (node 2). In case of errors of node 1, node 2 takes over the disks (like RAID or SSA disks) and the IP address (*hot standby*). When the failing machine becomes available it takes back the resources.
- Using a similar hardware configuration for hot standby, you may define *rotating standby*, which means that control and resources will not be returned to node 1 when it becomes available; it becomes the standby node.
- Both machines are doing work. Part of the disk pool is assigned to one, the remaining part is assigned to the other machine. Both are watching each other, ready to take over the others resources (*mutual operation takeover*).
- The disk pool is shared between both systems and can be accessed concurrently by both systems. Special daemons control and serialize competing disk access requests (*concurrent disk access*).

For more details about HACMP refer to the related AIX HACMP system documentation.

Since HACMP ensures availability of resources during system hardware or network failures and is independent of DCE, it is the ideal platform for the DCE security and CDS services. Both services can be replicated within DCE, but if the system which contains the master databases fails, write access is no longer possible. So there are good reasons to run Security Services and CDS in an HACMP cluster.

There are some rules which should be noticed:

- All information important for a certain DCE node must be configured into the same resource group. This includes the /var/dce, /etc/dce file systems and the IP address of the DCE node. It also may include file systems for the applications.
- Make sure that two different DCE machines are not migrated to one single machine in a failure situation.
- If you have a mutual takeover configuration, as described above, only one machine can be configured as DCE node.
- When configuring third-party takeover, only one of the production machines can be defined as DCE node.

Not only DCE core services can be made highly available with HACMP, any other DCE application can be implemented to HACMP. It may become expensive if applications are required to be available at many different locations. If a DCE application server needs to run on another platform than AIX, usage of other DCE dependent tools are required for replication.

A.3 SystemView and DCE

Several features of SystemView server for AIX require a DCE infrastructure for their communications:

- SNA Manager for AIX, which is a network manager
- Extended Systems Administration, which is a system management tool
- Printing Systems Manager based on the Palladium distributed print system (see 5.6.4, “Printing System Manager” on page 80)

Appendix B. DCE Software Ordering

This appendix is meant to help clarify which products and options should be specified when ordering DCE for various platforms. Note that the cell design should be complete when you begin ordering the software, many of the server products will only be necessary on a few of your nodes.

B.1 DCE Products for OS/2

For OS/2 Version 2 the following products are available:

5696-657 IBM DCE FOR OS/2 AND WINDOWS
5696-692 IBM DCE CLIENT FOR OS/2

For OS/2 Warp there will be a new product available which is currently in beta test. Its product number is

5622-851 IBM OS/2 WARP Server Version 4

The Directory and Security Server - OS/2 Warp Server provides a platform for the IBM OS/2 Directory and Security Server. This is IBM′s implementation of the DCE Cell Directory and Security Services. With this support, customers can implement an enterprise-wide directory and security model, based on open standards.

B.2 DCE Products for AIX

The current version of DCE for AIX 4.1 is Version 2 Release 1 which is a high performance implementation based upon DCE Version 1.1 from the Open Software Foundation (OSF).

B.2.1 DCE Client Software

With AIX 4.1 the client Software for DCE will be delivered together with the AIX base package.

B.2.2 DCE Server Software

When ordering DCE Software for AIX, you have to order the DCE products themselves as well as the related feature codes for the AIX product. The DCE Products are:

5765-532 GETTING STARTED WITH DCE
5765-533 AIX DCE SECURITY SERVICES
5765-534 AIX DCE CDS
5765-537 AIX DCE Enhanced DFS
5765-538 AIX DCE USER DATA MASKING
5765-540 AIX DCE NFS TO DFS Gateway

For the AIX 4.1 Program: 5692-AIX AIX VERSION 4 SPO the following feature codes should be considered:

0551 5765-533 DCE SECURITY SERVICES
0552 5765-534 DCE CELL DIRECTORY SERVICES
0553 5765-537 DCE ENHANCED DFS
0554 5765-540 DCE NFS TO DFS Gateway
0555 5765-532 GETTING STARTED WITH DCE

0556 5765-538 USER DATA MASKING

In March 1996, IBM announced the Directory and Security Servers for AIX V4 in the family of IBM Software Servers. Directory and Security Servers for AIX V4 (5765-639) includes:

- DCE Base Services
- DCE Security Services
- DCE Directory Services
- DCE Tools (IDL compiler, Sams utility, diagnostic tools)

B.2.3 DCE Application Software (optional)

CICS related Software:

5697-195 ENCINA MONITOR SUITE FOR AIX
5765-553 IBM CICS FOR AIX
5765-554 ENCINA CLIENT FOR AIX
5765-555 ENCINA PPC EXECUTIVE FOR AIX
5765-556 ENCINA STRUCTURED FILE SERVER AIX
5765-558 ENCINA SERVER FOR AIX
5765-559 ENCINA MONITOR FOR AIX

And the associated feature codes for the AIX 4.1 Program: 5692-AIX AIX VERSION 4 SPO

0547 5765-553 IBM CICS FOR AIX
0548 5765-559 ENCINA MONITOR
0549 5765-557 ENCINA PPC GATEWAY
0557 5697-195 ENCINA MON. SUITE
0558 5765-554 ENCINA CLIENT
0559 5765-558 ENCINA SERVER
0560 5765-556 ENCINA STR FILE SRV
0561 5765-555 ENCINA PPC EXEC
0595 5765-553 CICS INET GTW

In March 1996, IBM announced the IBM Transaction Server in the family of IBM Software Servers. IBM Transaction Server (5697-251) includes:

- CICS for AIX V2.1.1
- CICS Systems Manager for AIX V1.1.0
- CICS Clients V1.1.0
- ENCINA for AIX V2.2.0
- CICS Internet Gateway for AIX V1.1.0

Updates

Before installing the DCE software for AIX, contact the IBM Customer Service to get the latest updates for the ordered software

-or-

if You have access to the internet, use *FixDist* to retrieve the updates from the IBM FixDist-server.

IBM Printing System Manager (PSM) V1.2 for AIX (5765-273) is a printing administration product that requires a full DCE infrastructure.

Distributed Security Manager for AIX (Beta Program) allow a centralized administration of any DCE compliant platform.

B.3 DCE Products for OS/400

DCE Base Service Software for OS/400 Version 3 Release 6 can be ordered with the following product number:

5798-TBF DISTRIBUTED COMPUTING ENVIRONMENT BASE SERVICE FOR OS/400

The product will be available in 06/96.

B.4 DCE Products for VM/ESA

For VM/ESA, Version 2 Release 1.0 will provide the IBM OpenEdition Distributed Environment (DCE) feature (free of charge) to allow interoperation with other DCE platforms. The feature codes for the product

5654-030 VM/ESA Version 2

are

5989 VM/ESA V2 DCE Base Services
R0ASNC

In addition,

5735-FAC IBM TCP/IP V2 for IBM

is a prerequisite product and should be ordered.

B.5 DCE Products for OS/390

DCE components are shipped as base elements of OS/390. They require OpenEdition MVS elements to be installed as well.

DCE and OpenEdition elements of OS/390 are:

- OpenEdition MVS Services
- OpenEdition MVS Debugger
- OpenEdition MVS Shell & Utilities
- OpenEdition DCE Base Services (OSF DCE level 1.1)
- OpenEdition DCE Distributed File Service DFS (OSF DCE level 1.0.3a)

Optional elements of OS/390 are also available for encryption and security functions:

- OpenEdition DCE User Data Privacy
- OpenEdition DCE Security Server

OS/390 also contains as base elements the communication components such as TCP/IP V3 (5655-HAL) and VTAM V4R3 (5695-117) with the AnyNet/MVS feature.

IBM OpenEdition DCE Application Support for MVS/ESA Version 1 Release 2, 5655-064 is a separate product with three features:

- A CICS feature
- An IMS feature
- A CICS and IMS feature

Appendix C. Special Notices

This publication is intended to help network and application designers to determine DCE cell configurations. The information in this publication is not intended as the specification of any programming interfaces that are provided by DCE for OS/2, DCE for AIX, AS/400 DCE, IBM OpenEdition DCE for VM/ESA, and OpenEdition DCE for MVS/ESA. See the PUBLICATIONS section of the IBM Programming Announcements for more information about what publications are considered to be product documentation on the above DCE products.

References in this publication to IBM products, programs or services do not imply that IBM intends to make these available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that only IBM′s product, program, or service may be used. Any functionally equivalent program that does not infringe any of IBM′s intellectual property rights may be used instead of the IBM product, program or service.

Information in this book was developed in conjunction with use of the equipment specified, and is limited in application to those specific hardware and software products and levels.

IBM may have patents or pending patent applications covering subject matter in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to the IBM Director of Licensing, IBM Corporation, 500 Columbus Avenue, Thornwood, NY 10594 USA.

The information contained in this document has not been submitted to any formal IBM test and is distributed AS IS. The information about non-IBM (″vendor″) products in this manual has been supplied by the vendor and IBM assumes no responsibility for its accuracy or completeness. The use of this information or the implementation of any of these techniques is a customer responsibility and depends on the customer′s ability to evaluate and integrate them into the customer′s operational environment. While each item may have been reviewed by IBM for accuracy in a specific situation, there is no guarantee that the same or similar results will be obtained elsewhere. Customers attempting to adapt these techniques to their own environments do so at their own risk.

Reference to PTF numbers that have not been released through the normal distribution process does not imply general availability. The purpose of including these reference numbers is to alert IBM customers to specific information relative to the implementation of the PTF when it becomes available to each customer according to the normal IBM PTF distribution process.

The following terms are trademarks of the International Business Machines Corporation in the United States and/or other countries:

AIX	AIX/6000
AnyNet	AS/400
CICS	DB2
DFSMS	DRDA
IBM	IMS
LANDP	MQ
MQSeries	MVS/ESA
Open Blueprint	OpenEdition

OS/2	OS/390
OS/400	PS/2
RACF	RISC System/6000
RS/6000	S/390
SOMobjects	System Object Model
SystemView	VM/ESA
VTAM	400

The following terms are trademarks of other companies:

CA	Computer Associates
C + +	American Telephone and Telegraph Company, Incorporated
CORBA	Object Management Group, Incorporated
DCE	The Open Software Foundation
Encina	Transarc Corporation
HP	Hewlett-Packard Company
Macintosh	Apple Computer, Incorporated
Microsoft	Microsoft Corporation
NFS	Sun Microsystems Incorporated
Network File System	Sun Microsystems Incorporated
Novell	Novell, Incorporated
NT	Microsoft Corporation
Option	American Telephone and Telegraph Company, Incorporated
PC-NFS	Sun Microsystems Incorporated
OSF	Open Software Foundation, Incorporated
OSF/DCE	Open Software Foundation, Incorporated
Open Software Foundation	Open Software Foundation, Incorporated
POSIX	Institute of Electrical and Electronic Engineers
Sybase	Sybase Corporation
Transarc	Transarc Corporation
Visual Basic	Microsoft Corporation
Windows NT	Microsoft Corporation
Windows 95	Microsoft Corporation
X-Windows	Massachusetts Institute of Technology

C-bus is a trademark of Corollary, Inc.

PC Direct is a trademark of Ziff Communications Company and is used by IBM Corporation under license.

UNIX is a registered trademark in the United States and other countries licensed exclusively through X/Open Company Limited.

Microsoft, Windows, and the Windows 95 logo
are trademarks or registered trademarks of Microsoft Corporation.

Other trademarks are trademarks of their respective companies.

Appendix D. Related Publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this redbook.

D.1.1.1 General DCE Books

- *Understanding DCE Concepts*, GC09-1478
- *OSF/DCE User's Guide and Reference (Prentice Hall)*, SR28-4992
- *OSF/DCE Administration Guide: Core Components (Prentice Hall)*
- *OSF/DCE Administration Guide: Extended Services (Prentice Hall)*
- *OSF/DCE Administration Reference (Prentice Hall)*, SR28-4993
- *OSF/DCE Application Development Guide (Prentice Hall)*
- *OSF/DCE Application Development Reference (Prentice Hall)*, SR28-4995
- *Client/Server Survival Guide With OS/2 (Orfali and Harkey)*, SR28-5494
- *Distributed Computing: Implementation and Management Strategies (Prentice Hall)*, SR28-5709
- *OSF DCE Guide to Developing Distributed Applications (Lockhart)*, SR28-5853
- *OSF DCE A Guide to Developing Portable Applications (Peterson)*, SR28-5854
- *Understanding DCE, 2nd ed. (O'Reilly)*
- *Guide to Write DCE Applications, 2nd ed. (O'Reilly)*

D.1.1.2 OS/2 Platform

- *DCE Installation & User's Guide*, S64F-2269
- *DCE FOR OS/2 Installation Guide*, S96F-8502
- *DCE FOR OS/2 Administrator's Guide*, S96F-8504
- *DCE FOR OS/2: Administration Command Reference*, S96F-8505
- *DCE FOR OS/2: Application Development Guide*, S96F-8506
- *DCE FOR OS/2: Application Development Reference*, S96F-8507
- *DCE FOR OS/2 Master Index*, S96F-8615
- *DCE Windows Client Guide*, S96F-8622
- *DCE SDK for Windows Guide/Reference*, S96F-8623

D.1.1.3 AIX Platform

- *Introduction to DCE 2.1 for AIX*, SC23-2796
- *DCE V2.1 for AIX: Getting Started*, SC23-2797
- *AIX DCE GDS Administration Guide & Reference*, SC23-2602
- *HACMP for DCE and ENCINA Guide V1.3*, SC23-2737

The following documentation is available only to licensed users of the associated program products and will be supplied in softcopy format with the AIX Version 4.1.3 or higher operating system (form numbers are CD-ROMs):

- *Getting Started with DCE for Application Development V2.1.0*, LCD4-0142

- *DCE NFS to DFS Authenticating Gateway V2.1.0 for AIX*, LCD4-0141
- *DCE Cell Directory Services V2.1.0 for AIX*, LCD4-0139
- *DCE Security Services V2.1.0 for AIX*, LCD4-0138
- *DCE Enhanced DFS V2.1.0 for AIX*, LCD4-0140
- *DCE V2.1 Update Package for AIX V4.1*, LCD4-0199

D.1.1.4 OS/400 Platform

- *Introducing DCE/400*, GC09-1710
- *AS/400 DCE V3R1.0 Planning*, SC09-1711
- *AS/400 DCE V3R1.0 Configuring and Getting Started*, SC09-1712
- *AS/400 DCE V3R1.0 Application Development Guide*, SC09-1713
- *AS/400 DCE V3R1.0 Application Development Reference*, SC09-1714
- *AS/400 DCE V3R1.0 Administration Guide*, SC09-1715
- *AS/400 DCE V3R1.0 Administration Reference*, SC09-1716
- *Developing (REAL) DCE Applications for OS/400*, SG24-2572
- *AS/400 Distributed Computing Environment*, G325-6182

D.1.1.5 VM/ESA Platform

- *IBM OpenEdition DCE for VM/ESA: Administration Guide*, SC24-5730
- *IBM OpenEdition DCE for VM/ESA: Administration Reference*, SC24-5731
- *IBM OpenEdition DCE for VM/ESA: Application Development Guide*, SC24-5732
- *IBM OpenEdition DCE for VM/ESA: Application Development Reference*, SC24-5733
- *IBM OpenEdition DCE for VM/ESA: Configuring and Getting Started*, SC24-5734
- *IBM OpenEdition DCE for VM/ESA: Introducing the OpenEdition Distributed Computing Environment*, SC24-5735
- *IBM OpenEdition DCE for VM/ESA: Messages and Codes*, SC24-5736
- *IBM OpenEdition DCE for VM/ESA: Planning*, SC24-5737
- *IBM OpenEdition DCE for VM/ESA: User's Guide*, SC24-5738

D.1.1.6 OS/390 Platform

- *OpenEdition DCE Application Support for MVS/ESA, Programming Guide*, SC09-1530
- *OpenEdition DCE Application Support for MVS/ESA, Configuration and Administration Guide*, SC09-1659
- *OpenEdition DCE DFS Administration Guide and Reference*, SC28-1720
- *OpenEdition DCE DFS Configuration and Release*, SC28-1722
- *OS/390 OpenEdition DCE Planning*, SC28-1582
- *OS/390 OpenEdition DCE Configuring and Getting Started*, SC28-1583
- *OS/390 OpenEdition DCE Administration Guide*, SC28-1584
- *OS/390 OpenEdition DCE Command Reference*, SC28-1585

- *OS/390 OpenEdition DCE User's Guide*, SC28-1586
- *OS/390 OpenEdition DCE Application Development Guide: Introduction and Style*, SC28-1587
- *OS/390 OpenEdition DCE Apllication Development Guide: Core Components*, SC28-1588
- *OS/390 OpenEdition DCE Application Development Guide: Directory Services*, SC28-1589
- *OS/390 OpenEdition DCE Application Development Reference*, SC28-1590
- *OS/390 OpenEdition DCE Messages and Codes*, SC28-1591
- *RACF V2R2 Support for OS/390 OpenEdition DCE, SOMobjects*, GC23-3993

D.2 International Technical Support Organization Publications

For information on ordering these ITSO publications see "How To Get ITSO Redbooks" on page 103.

- *Developing DCE Applications for AIX, OS/2 & WINDOWS*, GG24-4090
- *OSF DCE for AIX, OS/2 & DOS WINDOWS Overview*, GG24-4144
- *The Distributed File System (DFS) for AIX/6000*, GG24-4255
- *Elements of Security: AIX 4.1*, GG24-4433
- *MVS/ESA OpenEdition DCE: Installation and Configuration Examples*, GG24-4480
- *MVS/ESA OpenEdition DCE: Application Develepment Cookbook*, GG24-4481
- *MVS/ESA OpenEdition DCE: Application Support Sersver CICS and IMS*, GG24-4482
- *RACF Support for Open Systems Technical Presentation Guide*, GG26-2005
- *Developing (REAL) DCE Applications for OS/400*, SG24-2572
- *VM/ESA OpenEdition DCE Introduction and Implementation Notebook*, SG24-4554
- *Understanding OSF DCE 1.1 for AIX and OS/2*, SG24-4616
- *MVS/ESA OpenEdition DCE: Application Support Servers CICS and IMS*, GG24-4482

A complete list of International Technical Support Organization publications, known as redbooks, with a brief description of each, may be found in:

International Technical Support Organization Bibliography of Redbooks, GG24-3070.

How To Get ITSO Redbooks

This section explains how both customers and IBM employees can find out about ITSO redbooks, CD-ROMs, workshops, and residencies. A form for ordering books and CD-ROMs is also provided.

This information was current at the time of publication, but is continually subject to change. The latest information may be found at URL http://www.redbooks.ibm.com/redbooks.

How IBM Employees Can Get ITSO Redbooks

Employees may request ITSO deliverables (redbooks, BookManager BOOKs, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **PUBORDER** — to order hardcopies in United States
- **GOPHER link to the Internet**

  Type GOPHER.WTSCPOK.ITSO.IBM.COM
- **Tools disks**

  To get LIST3820s of redbooks, type one of the following commands:

  ```
  TOOLS SENDTO EHONE4 TOOLS2 REDPRINT GET SG24xxxx PACKAGE
  TOOLS SENDTO CANVM2 TOOLS REDPRINT GET SG24xxxx PACKAGE (Canadian users only)
  ```

  To get lists of redbooks:

  ```
  TOOLS SENDTO WTSCPOK TOOLS REDBOOKS GET REDBOOKS CATALOG
  TOOLS SENDTO USDIST MKTTOOLS MKTTOOLS GET ITSOCAT TXT
  TOOLS SENDTO USDIST MKTTOOLS MKTTOOLS GET LISTSERV PACKAGE
  ```

  To register for information on workshops, residencies, and redbooks:

  ```
  TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ITSOREGI 1996
  ```

  For a list of product area specialists in the ITSO:

  ```
  TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ORGCARD PACKAGE
  ```
- **Redbooks Home Page on the World Wide Web**

  http://w3.itso.ibm.com/redbooks/redbooks.html
- **IBM Direct Publications Catalog on the World Wide Web**

  http://www.elink.ibmlink.ibm.com/pbl/pbl

  IBM employees may obtain LIST3820s of redbooks from this page.
- **ITSO4USA category on INEWS**
- **Online** — send orders to:

  USIB6FPL at IBMMAIL or DKIBMBSH at IBMMAIL
- **Internet Listserver**

  With an Internet E-mail address, anyone can subscribe to an IBM Announcement Listserver. To initiate the service, send an E-mail note to announce@webster.ibmlink.ibm.com with the keyword subscribe in the body of the note (leave the subject line blank). A category form and detailed instructions will be sent to you.

How Customers Can Get ITSO Redbooks

Customers may request ITSO deliverables (redbooks, BookManager BOOKs, and CD-ROMs) and information about redbooks, workshops, and residencies in the following ways:

- **Online Orders** (Do not send credit card information over the Internet)

  IBMMAIL — send orders to:

In United States:	usib6fpl at ibmmail
In Canada:	caibmbkz at ibmmail
Outside North America:	bookshop at dkibmbsh at ibmmail

  Internet — send orders to:

In United States:	usib6fpl@ibmmail.com
In Canada:	lmannix@vnet.ibm.com
Outside North America:	bookshop@dk.ibm.com

- **Telephone orders**

United States (toll free)	1-800-879-2755
Canada (toll free)	1-800-IBM-4YOU
Outside North America	(long distance charges apply)
(+45) 4810-1320 - Danish	(+45) 4810-1020 - German
(+45) 4810-1420 - Dutch	(+45) 4810-1620 - Italian
(+45) 4810-1540 - English	(+45) 4810-1270 - Norwegian
(+45) 4810-1670 - Finnish	(+45) 4810-1120 - Spanish
(+45) 4810-1220 - French	(+45) 4810-1170 - Swedish

- **Mail Orders** — send orders to:

IBM Publications Publications Customer Support P.O. Box 29554 Raleigh, NC 27626-0570 USA	IBM Publications 144-4th Avenue, S.W. Calgary, Alberta T2P 3N5 Canada	IBM Direct Services Sortemosevej 21 DK-3450 Allerød Denmark

- **Fax** — send orders to:

United States (toll free)	1-800-445-9269
Canada (toll free)	1-800-267-4455
Outside North America (long distance charge)	(+45) 48 14 2207

- **1-800-IBM-4FAX (United States)** or **(+1) 415 855 43 29 (Outside USA)**

  Ask for:
  Index # 4421 Abstracts of new redbooks
  Index # 4422 IBM redbooks
  Index # 4420 Redbooks for last six months

- **Direct Services**

  Send note to softwareshop@vnet.ibm.com

- **Redbooks Home Page on the World Wide Web**

  http://www.redbooks.ibm.com/redbooks

- **IBM Direct Publications Catalog on the World Wide Web**

  http://www.elink.ibmlink.ibm.com/pbl/pbl

- **Internet Listserver**

  With an Internet E-mail address, anyone can subscribe to an IBM Announcement Listserver. To initiate the service, send an E-mail note to announce@webster.ibmlink.ibm.com with the keyword subscribe in the body of the note (leave the subject line blank).

IBM Redbook Order Form

Please send me the following:

Title	Order Number	Quantity

- **Please put me on the mailing list for updated versions of the IBM Redbook Catalog.**

First name Last name

Company

Address

City Postal code Country

Telephone number Telefax number VAT number

- Invoice to customer number

- Credit card number

Credit card expiration date Card issued to Signature

We accept American Express, Diners, Eurocard, Master Card, and Visa. Payment by credit card not available in all countries. Signature mandatory for credit card payment.

DO NOT SEND CREDIT CARD INFORMATION OVER THE INTERNET.

List of Abbreviations

ACL	Access Control List
AIX	Advanced Interactive Executive
ANSI	American National Standards Institute
API	Application Programming Interface
APPC	Advanced Programming to Programming Communication
ASCII	American National Standard Code for Information Interchange
ATM	Asynchronous Transfer Mode
CCITT	International Consultative Committee on Telegraph and Telephone
CD	Compact Disk
CD-ROM	Compact Disk/Read-Only Memory
CDS	Cell Directory Service
CTS	Common Transport Semantic
CICS	Customer Information Control System
CORBA	Common Objects Request Broker Architecture
DB2	Database 2 (Relational Database Management System)
DBMS	Database Management System
DACM	Distributed Access Control Manager
DCE	Distributed Cell Environment
DDKT	Doxa Distribution ToolKit
DFS	Distributed File System
DNS	Domain Name Service
DOS	Disk Operating System
DRDA	Distributed relational Database Architecture
DSA	Directory System Agent
DSM	Distributed Security Manager
DSOM	Distributed System Object Model
DSSM	Distributed System Services
DTS	Distributed Tine Service
DUA	Directory User Agent
ENCINA	Enterprise Computing in a New Age (TransArc transaction monitor for RS/6000)
ESA	Extended System Architecture
FBSS	Financial Branch System Service
FDDI	Fiber Distributed Data Interface
FLDB	File Location Database
GDA	Global Directory Agent
GDS	Global Directory Services
GUI	Graphical User Interface
HACMP	High Availability Cluster Multi-Processing
HFS	Hierarchical File System
HP	Hewlett-Packard Co.
HPFS	High Performance File System
IBM	International Business Machines
IDL	Interface Definition Language
IEEE	Institute of Electrical and Electronical Engineers
IMS	Information Management System
IP	Internet Protocol
IPF/X	Interactive Productivity Facility for X-Windows
ISO	International Organization for Standardization
IT	Information Technology
ITSO	International Technical Support Organization
JAD	Joint Application Point
LAN	Local Are Network
LANDP	LAN Distributed Platform
LFS	Local File System
MPTS	Multi-Protocol Transport Service
MPTN	Multi-Protocol Transport Networking
MVS	Multiple Virtual Storage
MQ	Message Queuing

NFS	Network File System
NTP	Network Time Protocol
NT	Microsoft Windows NT (New Technology)
OE	OpenEdition
OEC	Open Environment Corporation
OEM	Original Equipment Manufacturer
OMG	Object Management Group
OS/2	Operating System 2
OS/390	Open Server/390
OS	Operating System
OSA	Open System Adapter
OSF	Open Software Foundation
PAC	Privilege Access Certificate
PDS	Partitioned Data Set
POSIX	Portable Operating System Interface for computer environments, an IEEE operating system standard closely related to the UNIX system
PS/2	Personal System 2
PSM	Print Management System
PTF	Program Temporary Fix
RACF	Resource Access Control Facility
RAID	Redundant Array of Independent Disks
RISC	Reduced Instruction Set Computer
RPC	Remote Procedure Call
RS/6000	IBM RISC System/6000
SAF	Security Authorization Facility
SDK	System Developers Kit
SMIT	System Management Interface Tool
SNA	System Network Architecture
SOM	System Object Model
SSA	Service Support Adapter
TCP	Transaction Control Protocol
TP	Teleprocessing
TSO	Time Sharing Option
UNIX	UNIX is a registered trademark in the United States
UTC	Universal Time, Coordinated
VM	Virtual Machine
VTAM	Virtual Telecommunication Access Method
WAN	Wide Are Network
WARP	Workstation Asset Reduction Program

Index

Special Characters

A

B

C

K

L

M

N

O

P

Q

R

S

T

V

X

Printed in U.S.A.