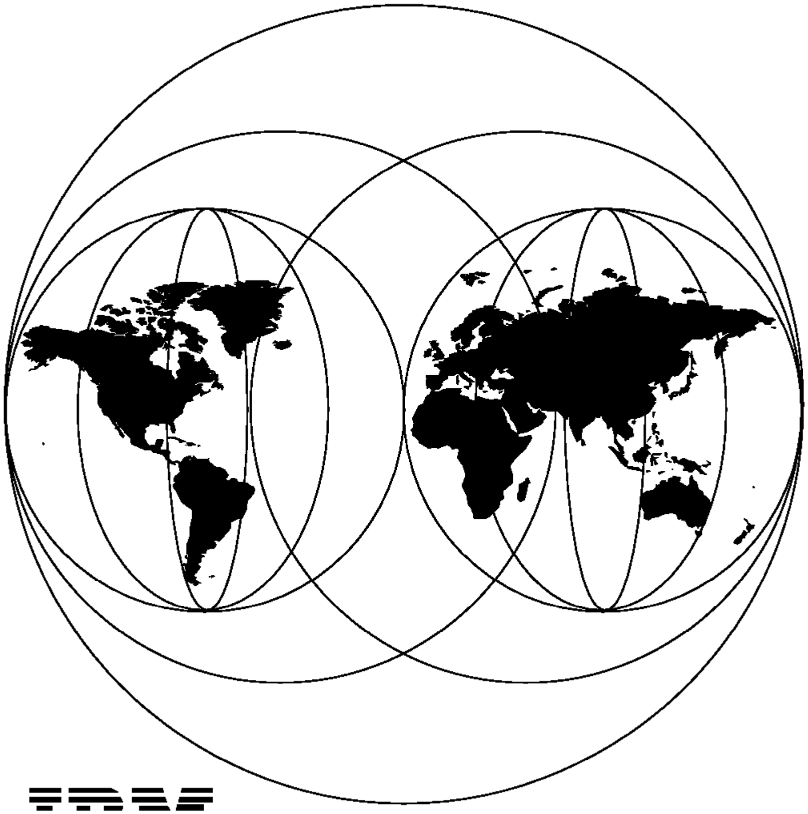
SG24-2543-01

IBM DSS and DCE Cross-Platform Guide

December 1996



International Technical Support Organization
Austin Center

International Technical Support Organization

SG24-2543-01

IBM DSS and DCE Cross-Platform Guide

December 1996

Take Note!

Before using this information and the product it supports, be sure to read the general information in Appendix B, “Special Notices” on page 155.

Second Edition (December 1996)

This edition applies to:

- OpenEdition DCE for use with OS/390
- OpenEdition DCE for VM/ESA, a feature of VM/ESA Version 2
- DCE Base Serivces/400 Version 3 for use with OS/400 Version 3
- DSS and DFS for AIX Version 4 Product Family for use with AIX Version 4
- DSS and DFS for OS/2 Warp, Version 4
- DCE for Windows 3.1 and Windows NT
- Other IBM and non-IBM products

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. JN9B Building 045 Internal Zip 2834
11400 Burnet Road
Austin, Texas 78758-3493

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

The OSF DCE has been gaining a lot of attention across all IBM platforms and products since IBM endorsed this technology for the Open Blueprint. It is the strategic platform for distributed services, and many existing products have already been, or will be, integrated with DCE services.

This document describes IBM′s current and future cross-platform DCE offerings and also mentions important vendor products based on OSF DCE. It discusses the operating system environments and DCE implementations on all IBM platforms as well as user integration, administration, and application development. Also, current products from IBM and some third-party software exploiting DCE technology are listed and explained. Furthermore, it gives an outlook on future trends and directions and provides useful sources for more information.

Customers, system engineers, and marketing representatives will also gain a better understanding of all IBM platforms and their DCE implementation specifics, as well as of IBM's strategic products which are based on DCE.

How This Redbook Is Organized

The document is organized as follows:

- **Chapter 1, "Distributed Computing and IBM′s Strategy" on page 1**

  This chapter explains the requirements for open distributed computing, the strengths of DCE and IBM′s strategy, the Open Blueprint. It also discusses what this means for existing or future IBM products.

- **Chapter 2, "DCE Overview" on page 11**

  This chapter is intended for anyone who needs to understand the basic DCE components and how they work. Readers with DCE experience can skip it.

- **Chapter 3, "DCE Implementations on IBM Platforms" on page 21**

  This chapter explains, per IBM platform, the operating system environment, such as threads and the file system, as well as the DCE implementation, packaging and order numbers, the user environment, administration, and application development.

- **Chapter 4, "DCE Implementations on Non-IBM Platforms" on page 81**

  This chapter lists and shortly explains implementations of DCE on some non-IBM platforms.

- **Chapter 5, "Security in a Distributed Environment" on page 87**

  Security is a strong requirement in a distributed environment. This chapter explains products, programming interfaces, and concepts that are available to provide enterprise-wide integrated security.

- **Chapter 6, "IBM′s Software Products Will Exploit DCE" on page 91**

  IBM′s commitment to DCE in the Open Blueprint means that existing products will be changed over time to incorporate these layers and services. This chapter introduces several IBM products that will be the first to utilize DCE services.

- **Chapter 7, “Client/Server Application Development” on page 107**

  This chapter discusses several approaches and tools to develop distributed applications that make use of DCE.

- **Chapter 8, “DCE Administration Tools” on page 129**

  In this chapter, we give a platform-independent view of DCE administration with standard and nonstandard tools.

- **Chapter 9, “DCE Evolution and Future Directions” on page 135**

  DCE is still evolving, gaining new features, improving performance, and integrating more smoothly into existing information systems. This chapter covers future directions of OSF DCE.

- **Appendix A, “Other Sources of Information” on page 147**

  This appendix describes where and how to learn more about DCE. A lot of information is produced by the OSF and the companies involved in DCE, including IBM and Transarc Corp. Most of it is available in electronic form on the Internet.

- **Appendix B, “Special Notices” on page 155**

  This appendix contains the terms that apply for this publication and a list of trademarks.

- **Appendix C, “Related Publications” on page 159**

  This appendix lists publications from IBM and from other sources pertinent to Distributed Computing Environment and its implementations on various platforms.

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 Austin Center or providing support from their work location:

Heinz Johner is an Advisory Systems Engineer at the International Technical Support Organization, Austin Center. Before joining the ITSO in mid-1996, he worked for IBM Switzerland, and he had oversight responsibility for AIX, DCE, and systems management projects at large customers. He was also involved as a consultant in various other customer projects in the same technical areas.

Bjarne Rasmussen is a Senior Systems Engineer at the Nordic Open Systems Center in Denmark. He is doing consultancy for large customers in the Nordic countries, and his area of expertise is interoperability based on DCE, Internet, Lotus, and Microsoft standards and tools.

Scott Vetter is an ITSO VM Area Specialist in the United States. His areas of expertise include OpenEdition functionality covering the POSIX, DCE, and GUI components. He has written many redbooks that cover DCE, POSIX, GUI, and core VM functionality.

We would also like to acknowledge the professionals who took the time to review this publication and provided invaluable help and advice:

Oscar Cepeda	IBM Austin, ITSO Center
Steve Dahlby	IBM Rochester (OS/400 Development)
Hilding Landen	IBM Sweden
Peter S. Wassel	IBM U.S., OS/390 DCE Development
Michael J. Morton	IBM U.S., OS/390 DCE Development
Luis Casco-Arias	IBM Austin
Donna Barker	IBM Austin
John McMeen	IBM Austin

The authors of the first release of this document, *IBM DCE Cross-Platform Guide*, GG24-2543-00, were:

Rolf Lendenmann (Leader)	IBM ITSO Austin
Pascal Meheut	IBM France
Jener Takeshi Sato	Itec Brazil
Scott Vetter	IBM ITSO Poughkeepsie (VM/ESA)

Advice and assistance in the reviewing of the first release of this document came from:

John Milbum	IBM Austin
Donna Barker	IBM Austin
Donna Edgar	IBM Austin
Ed Tidik	IBM Printing Systems Co.
Brian Smith	IBM ITSO Rochester (AS/400)
Rich Conway	IBM ITSO Poughkeepsie (MVS/ESA)
Stephen Breese	IBM ITSO Raleigh (LANDP)
Rob Macgregor	IBM ITSO Raleigh (NetSP)
Salvatore La Pietra	Security Center of Competence, IBM Germany

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. Distributed Computing and IBM′s Strategy

In search of higher profitability, better service to their customers, and reduced expenses, businesses are reviewing their fundamental operations with special emphasis on their investments in information technology (IT). This chapter explains why customers are moving to distributed computing and the solutions that are in place today to serve the particular needs of single departments or user groups within a company. After listing requirements for global enterprise solutions, we show what DCE can do to tie everything together and build an excellent basis to fulfill all these requirements. Eventually, we discuss how IBM incorporates the DCE technology into its strategy for open distributed computing, the Open Blueprint, and what this means for IBM′s product set.

1.1 The History of Distributed Computing

For historic reasons, many companies still have centralized computing with large mainframes. Business-critical applications run in this environment, which still has advantages, such as low cost per user, high availability, single point of administration, and so forth, over a client/server environment. However, in many cases, this centralized mainframe model no longer matches the structure of a company and the way it′s doing business.

Companies are reengineering their businesses and splitting them up into smaller units that are easier to manage, responsible for their own operations and profit, and more responsive to the needs of the market. The IT operations need to serve the business needs of a company; so they need to be reengineered as well. The computing power should be as close to the end user as possible; solutions should be flexible, scalable, and portable to different platforms, while access to common data and certain mainframe applications must still be maintained.

It wasn′t just business process reengineering that brought about the movement toward decentralized computing. Sophisticated programs with attractive graphical user interfaces that were available on personal computers (PCs), the falling prices for PCs, and the ability to connect them in local area networks (LANs) made departments of bigger companies buy PCs and LANs for purposes that were independent of the mainframes. Communication protocols and software were implemented to provide access to common data (file transfer) and programs (remote login and terminal emulation) residing on the corporate mainframe level. The number of PCs grew, and the PC applications soon required more resources and more powerful machines as they became more sophisticated. The administration also became more and more expensive. This led to connecting the PCs and having them share resources. Products such as IBM LAN Server, Novell NetWare, or Microsoft LAN Manager were installed; more powerful UNIX workstations and servers using file sharing systems, such as Sun Network File System (NFS), were used; remote procedure calls (RPC) for two-way communications and cooperative processing were employed.

What we see are needs for new distributed computing solutions coming from two different sides:

- From top down. Reengineering business processes usually means down-sizing or right-sizing to open distributed platforms while not sacrificing the advantages of the mainframe environment.

- From the bottom up. The heterogeneous, often proprietary, local solutions should be turned into reliable, secure, and global solutions.

1.2 Requirements for Global Distributed Solutions

From the discussion in 1.1, “The History of Distributed Computing” on page 1 above, the requirements for global distributed solutions can be derived and summarized as follows:

- Transparent Global Access

  Users need access to data and applications wherever they reside. This must be transparent so that people are not forced to navigate over physical barriers or boundaries and have to know the physical IT infrastructure.

- Interoperability

  Products from all suppliers must be able to work together. Openness, achieved by the use of formal and informal standard interfaces, formats, and protocols in products, will facilitate interoperability in a multivendor, heterogeneous environment.

- Application Portability

  Customers will choose applications, and they will want to pick the hardware platform independently to retain flexibility and the ability to switch. They may then choose other platforms for development or administration depending on the available tools.

  Developers need tools to design, build, test, and maintain *reusable* applications in the most efficient way.

- Manageability

  System administrators need tools to manage users, data, applications, security, and to provide network monitoring or problem determination.

- Investment Protection

  No company can afford to ignore previous investments and start from scratch. Therefore, open distributed implementations must include current (sometimes called legacy) applications and operating environments.

- Scalability

  Solutions must be flexible and must be able to accommodate a growing number of users or applications by adding, rather than replacing, resources. Reconfiguration for load balancing should be possible and easy to perform.

- Availability

  Applications must be highly available and able to survive a failure of hardware components.

- Security

  The global environment must be secure. Communication links (LANs and WANs) can be tampered with. Furthermore, resources must be protected with fine granular permissions and access control, and attempts to perform certain security relevant actions must be logged and audited.

1.3 The Distributed Computing Environment

While many of the solutions mentioned in 1.1, “The History of Distributed Computing” on page 1 are well established, they cannot fulfill *all* of the requirements listed in 1.2, “Requirements for Global Distributed Solutions” on page 2. Some have tight administrative and operative boundaries, thus lacking global transparency; others may have weak security, are not scalable, are not platform independent, cannot provide load balancing or high availability, and so forth. The Open Software Foundation (OSF) Distributed Computing Environment provides all the necessary features and services to *create* solutions and applications that satisfy *all* requirements.

The mission of the OSF′s Distributed Computing Environment is to provide a common infrastructure for distributed processing. By making use of the standard interfaces provided by DCE, applications are interoperable with, and portable to, other DCE platforms. All this despite the fact that those other platforms may otherwise be very different in terms of hardware, operating system, network transport, and application software. DCE primarily addresses:

- Client/server definition (function shipping)

  Distributed applications require a means by which to invoke remote computation. The dominant structuring paradigm today is that of client/server, wherein a server application exports a particular function or service to the client application. This paradigm promotes clear interfaces between clients and servers, thus easing the interconnection of many clients to a single server and a single client to many servers.

- Resource location

  Within a client/server organization, it is often desirable to relocate servers or to replicate servers for availability. This makes it necessary that specific endpoints (network addresses) not be included within application code. Rather, clients provide the name of the services they require, and a name service determines the address of an available instance of that server.

- Security

  Distributed applications must offer sufficient levels of security so that access to system resources can be appropriately protected. In particular, it is necessary to support authentication (the identification of individual participants), authorization (the restriction of services to those clients with the proper capabilities), integrity (the certainty that messages arrive unchanged), and privacy (by means of data encryption).

- Scalability

  With the advance of worldwide networks, such as the Internet, it is vital that distributed systems scale to support large local- and wide-area networks. This scaling must be accomplished in a manner that offers global connectivity and adequate performance without sacrificing local control over configuration. DCE offers the concept of a cell for this purpose. Cells are independent administrative domains that support secure intra-cell and inter-cell access.

- Resource sharing

  DCE Distributed File System (DFS) provides an advanced shared file system where users and applications may access files stored remotely in the same manner that they are accessed locally. Advanced caching schemes ensure that performance is not sacrificed and that local POSIX-sharing semantics are applicable to the distributed file system.

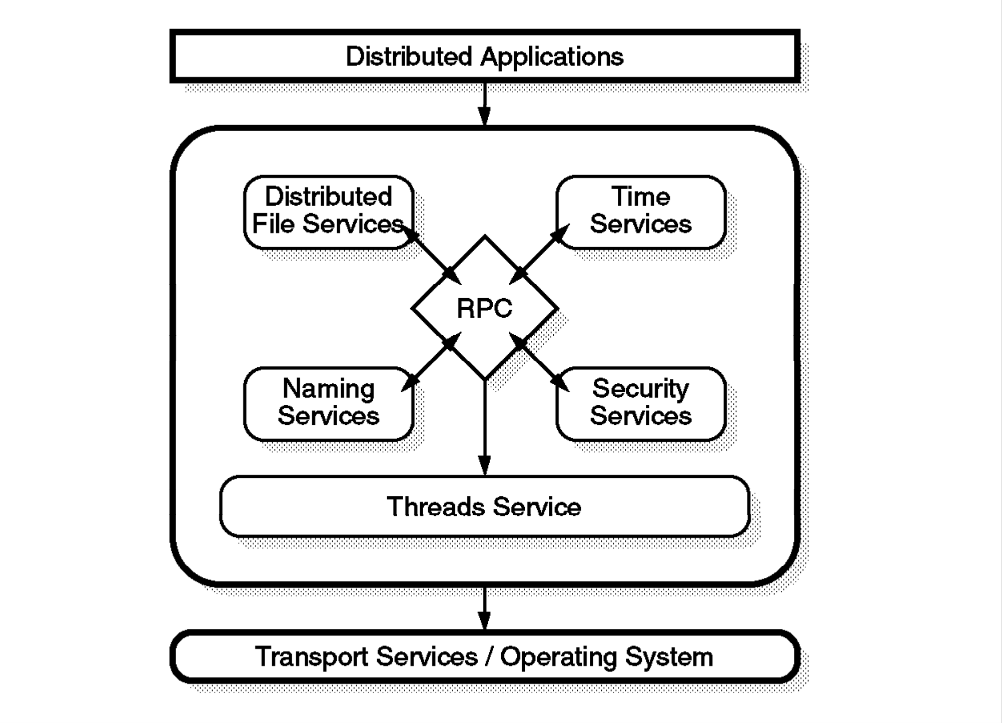


Figure 1. OSF Distributed Computing Environment

At the core of DCE is support for client/server communication via the remote procedure call (RPC). The rationale for RPC is simply to package remote units of computation the same way local ones are packaged within a procedure definition. RPC is the natural choice for client/server paradigm, and it simplifies the development of distributed applications by providing a formal specification of the interface between clients and servers. Programmers on either side are then free to code to that specification without concern for managing connections, message packing and unpacking, and the implicit shared state required by more complex communication paradigms.

RPC is synchronous, which means that the client typically blocks while waiting for a response from the server (although DCE does support one-way RPCs). While this greatly simplifies application development, clients must not be precluded from accessing multiple servers simultaneously. Thus, also fundamental to DCE, there is support for multithreading, whereby multiple units of computation (flows of control) may be active within a single address space. Threads provide a means for concurrency within applications and for efficient replication of services. The former allows a single client to carry on many RPC conversations at once, while the latter permits a single server to simultaneously handle multiple client requests.

Unlike many RPC mechanisms, the DCE RPC supports authentication via MIT′s Kerberos technology. By authenticating RPCs, participating clients and servers are each assured of the identity of its counterpart. DCE RPC also provides integrity and privacy of the messages as a countermeasure against threats, such as unwanted data modification and data eavesdropping.

The components of the Distributed Computing Environment are illustrated within Figure 1. In addition to the components that have been explicitly mentioned, DCE includes a name service for dynamically locating servers in the environment, a security service supporting authentication, authorization and privacy, a distributed time service for addressing discrepancies between remote clocks, and a user/group/account registry for managing access control lists.

However, the core DCE is basically a development environment for distributed applications. As such, it is extremely powerful with its replicatable services. Customers do not buy DCE just for the sake of it. They need applications that exploit all the powerful features of DCE. The DFS is such an application, a file sharing application. Many more applications are already available. The Open Group and OSF maintain databases and catalogs with applications using DCE technology (see A.3, “The Open Software Registry” on page 151 and A.4, “The OSF DCE Product Catalog” on page 151). Moreover, IBM has defined its strategy for distributed computing in the Open Blueprint. One of the building blocks is DCE. This means that many more existing and new IBM products will get DCE integration. For a more detailed description of the DCE architecture, you can consult Chapter 2, “DCE Overview” on page 11.

1.4 The IBM Open Blueprint

The Open Blueprint is a standards-based architecture that defines the services required by applications in a distributed multivendor environment.

The Open Blueprint can be viewed on three levels: as a world view, as a set of technologies, and as a basis for specific products.

The picture representing an architecture is often drawn as a set of related blocks, each containing a function that is important to that view. The Open Blueprint is made up of blocks, each representing functions that provide the services required by applications in a heterogeneous distributed environment. The arrangement of the blocks is important, as are the spaces between them. Sometimes, the blocks appear to form layers. In those cases, more explanation than just the picture is necessary to determine the relationships between the layers. An architecture is rigidly layered if the functions in the top layers are dependent on the lower-layered functions and must use each layer between them and a target function to access the target function. An architecture is functionally-layered if the functional dependencies are not necessarily vertical, and there is no requirement to drill down through the layers to get at a function in a lower layer. The Open Blueprint is functionally-layered.

Once the perspective is known, and the functions or the blocks in the diagram are specified, the second level involves the nature of the technologies for each function or building block in the architecture. The choices are key because they indicate the degree of openness and interoperability inherent in the architecture. The technologies included in the Open Blueprint, such as the Open Software Foundation′s Distributed Computing Environment (DCE) and the Object Management Group′s Common Object Request Broker Architecture (CORBA) and Common Object Services Specification (COS) embrace open standards.

Finally, there are the products that implement and use the functions defined in the architecture. There are many IBM and non-IBM products that deliver Open Blueprint function today, and more will come. There is not necessarily a one-to-one relationship between a block in the Open Blueprint and a product.

This is because there are other than architectural considerations for the contents or products. For example, if certain functions are always performed together, it might make sense to package them together in one product.

The Open Blueprint addresses the challenges of the open environment by viewing a system as part of a distributed network and viewing the network as if it were a single system.

The Open Blueprint enables a network of operating systems to function as a unit, as a ″network operating system″. A network operating system is comprised of multiple systems separated from each other and connected by a communication network. Services on each system cooperatively manage resources across the network in the same way a single operating system manages resources on one platform. Each node in the network can be thought of as structured according to the Open Blueprint. While all services need not be on all platforms, all services need to be accessible by applications on all platforms. The equivalent services on each platform work together to provide seamless, network-wide support for distributed and client/server applications.

Figure 2 on page 7 represents one instance of a platform in the network. The true distributed nature of the Open Blueprint is a network in which each individual system can be thought of as structured according to the Open Blueprint.

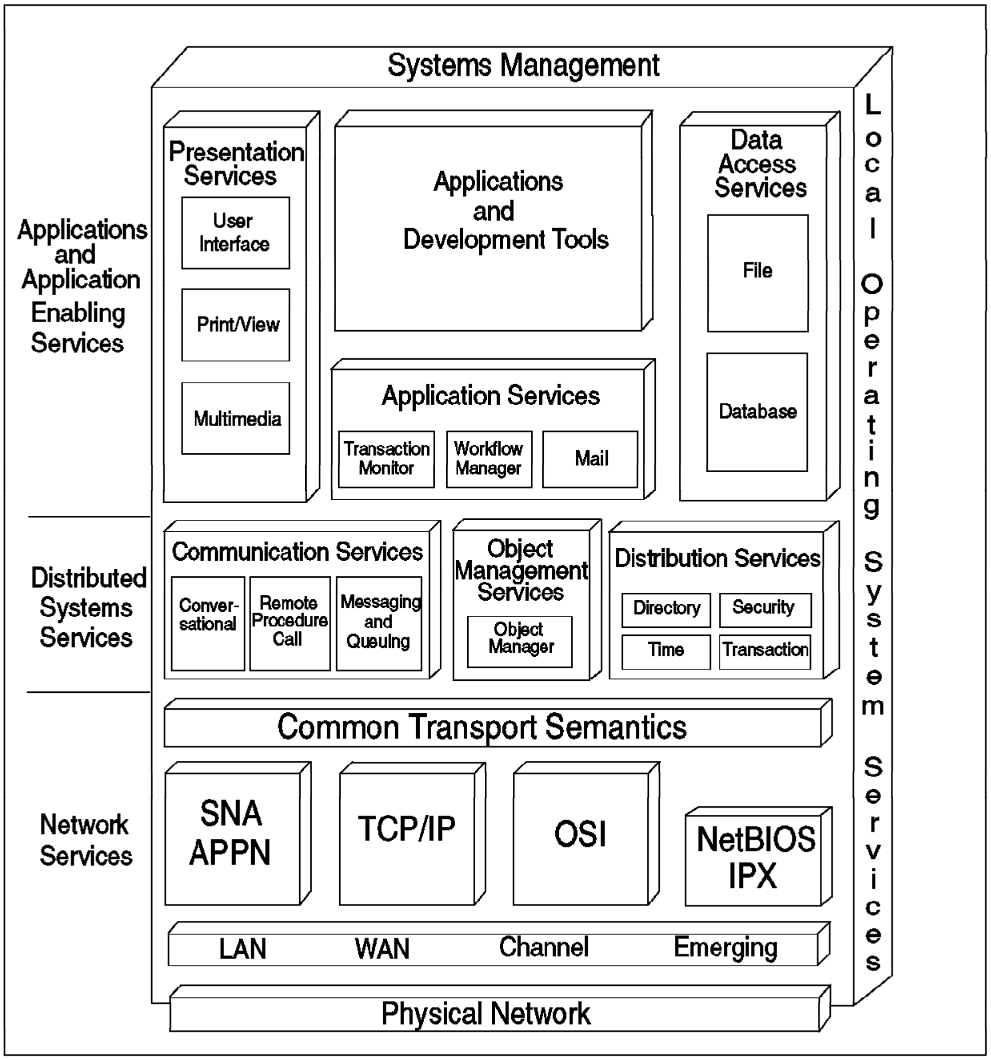


Figure 2. The Open Blueprint

As shown in Figure 2, there are several sets of resource-management services in the Open Blueprint.

Network Services

- Common Transport Semantics supports protocol-independent communication in distributed networks.
- Transport Services provides the protocols for transporting information from one system to another, such as SNA/APPN, TCP/IP, OSI, NetBIOS, and IPX.
- Signalling and Control Plane provides the ability to establish subnetwork specific connections.
- Subnetworking provides functions dealing with specific transmission facilities, such as various kinds of LANs, WANs, channels, Asynchronous Transfer Mode (ATM), and emerging technologies such as wireless networks.

Distributed Systems Services

- Distribution Services assist the communication between parts of distributed applications and resource managers by providing common functions such as a directory, security, and time. Distributed Services is based on Distributed Computing Environment (DCE) services from The Open Group. DCE has the functionality needed at this layer, and DCE is based on open standards. This is important for IBM and its customers.

- Communication Services provide mechanisms for parts of a distributed application or resource manager to talk to each other. Remote Procedure Call (RPC), which is also a DCE component, is included in the Communication Services.
- Object Management Services provide common object services, including transparent access to local and remote objects.

Application Enabling Services

- Presentation Services define the interaction between applications and the user.
- Application/Workgroup Services are common functions, such as mail, which are available for use by all applications.
- Data Access Services allow applications and resource managers to interact with various types of data.

The Open Blueprint provides guidelines for the integration of multivendor systems and the simplification of the more cumbersome aspects of distributed computing, such as multiple logins, multiple passwords, and unique application directories for locating resources. Products that align with the Open Blueprint provide designated interfaces and protocols. Products that use resource managers defined within the Open Blueprint, rather than their own unique mechanisms, are truly integrated with the Open Blueprint. For example, if every application uses its own unique security facility, a user must present a unique password for each application. If applications and resource managers use the security services from the Open Blueprint, a single user password will suffice for the cooperating applications. This integration improves the single-system image of the distributed system as perceived by the end user and application developer.

When considered as an architecture or world view, the Open Blueprint and its component blocks represent abstract functions or services that create a robust distributed environment. At this level, the question of whether it represents a ″procedural″ or ″object″ view is not relevant. It is only at the next level, the technologies chosen to implement the functions, that the distinction is visible. The Open Blueprint includes both procedural and object-oriented interfaces and protocols from IBM and other industry sources. Those from industry sources are broadly accepted standards. Many of these standards are the same ones included in X/Open′s Distributed Computing Services (XDCS) framework and OMGs Common Object Services specifications (COS).

The blocks in the Open Blueprint diagram, Figure 2 on page 7, do not correspond to specific products. The Open Blueprint is implemented by different products on different system platforms. Also, the Open Blueprint does not describe how the implementing software is packaged into offerings. Rather, the Open Blueprint describes the technical attributes and characteristics of supporting software, reflects desirable functional modularity, provides software principles and guidelines, and specifies important boundaries and interfaces.

The Open Blueprint describes techniques for building an open, heterogeneous, distributed system that is extensible by alternate component implementation and by support for evolving new technologies and functions. For example, the Open Blueprint currently supports three models for interprocess communication: Conversational, Remote Procedure Call (RPC), and Messaging and Queuing. If a fourth model were to be developed, it would be evaluated for inclusion. Because the Open Blueprint is modular, it is obvious that the fourth model would fit in next to the other three.

The Open Blueprint is a structure for distributed computing that will help IBM and others to deliver integrated, interoperable products and solutions:

- **For end users**, Open Blueprint integration hides the complexities of the network and makes it appear as a single system.
- **For application developers**, standard interfaces enable a single system view of the network and allow for the development of interoperable applications that can run on many platforms.
- **For system administrators**, the Open Blueprint defines a consistent way to manage the network to hide the complexities from application developers and end users.

IBM has elected DCE functions and technology for the corresponding functional blocks in the Open Blueprint. Due to this commitment, IBM offers DCE on all of its platforms and will continue to integrate DCE technology into its middleware software products and solutions.

1.5 IBM DSS and DCE Products

It is the purpose of this document to describe the various implementations of IBM's products which implement OSF DCE technology. Although all of them incorporate basic DCE technology from OSF and provide DCE core services to applications, recent developments from IBM added additional functions and tools and even integrated DCE services in other middleware products. It is therefore not adequate to just name these products "DCE". Recently announced products, incorporating DCE core services and IBM add-ons, are therefore called *Directory and Security Server* (DSS) rather than "DCE". An example is the IBM Directory and Security Server for OS/2Warp, which integrates DCE services with the OS/2 Warp Server or the OS/2 LAN Server.

Throughout this booklet and in the literature, the term *DCE Secure Core Services* (or even only *DCE Core Services*) is often used to describe the minimal DCE functions usually implemented on a certain platform:

- Threads
- Remote Procedure Call (RPC)
- Security Client
- Directory Client
- Time Services Client (not always included)

IBM has DCE implementations on all major platforms. They are described in Chapter 3, "DCE Implementations on IBM Platforms" on page 21.

Chapter 2. DCE Overview

DCE is establishing a defacto standard in the client/server arena and is supported, or at least announced, on major IBM platforms, such as AIX, OS/2 Warp, MVS, VM, and OS/400. It is also available on major competitor platforms, such as Microsoft Windows, DEC VMS, Siemens-Nixdorf BS/2000, HP MPE/ix, and on all the UNIX flavors including: AIX, OSF/1, Solaris, HP/UX, DG/UX, Sinix, and SCO. After the announcement of the merge between UNIX International and OSF members, it has become clear that DCE is going to be adopted in the NIS/NFS community, too.

This chapter explains what DCE is, although this information can be found in almost every book about DCE over and over again. If you know what the DCE components are and do, skip this chapter. However, since the target audience includes specialists of platforms on which the DCE technology is new, we decided it would be handy to find a basic introduction in this book, too.

2.1 OSF DCE Architecture

OSF DCE is a complete architecture that takes full advantage of the client/server paradigm. It offers a set of services and APIs, that can be used to build distributed applications and a set of management tools to manage the distributed environment. It can interoperate with other environments.

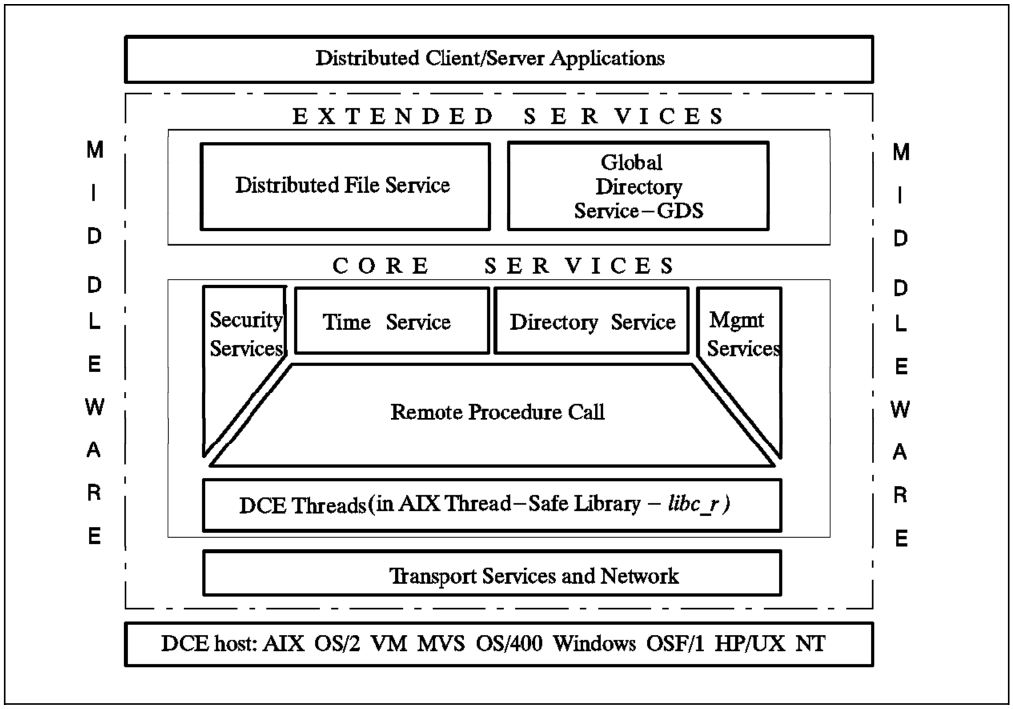

Distributed Client/Server Applications
EXTENDED SERVICES
MIDDLEWARE
MIDDLEWARE
Distributed File Service
Global Directory Service−GDS
CORE SERVICES
Security Services
Time Service
Directory Service
Mgmt Services
Remote Procedure Call
DCE Threads(in AIX Thread−Safe Library − libc_r)
Transport Services and Network
DCE host: AIX OS/2 VM MVS OS/400 Windows OSF/1 HP/UX NT

Figure 3. DCE Architecture

If we consider DCE as *middleware* (as is the case with most of the network operating systems, such as Novell NetWare), the operating system is hidden to a certain extent by a set of core and extended services offered by DCE. The users will see only the distributed client/server application. It will be completely transparent to them whether the application is local or distributed and no matter what operating system is underneath a distributed service. The architecture viewing DCE as middleware is explained in the following sections.

2.2 OSF DCE Components and Services

The following sections provide a short description of the DCE components and technology. At the end of each section, you will be referred to other documentation that contains a full description.

2.2.1 DCE Threads

Threads support the creation, management, and synchronization of multiple concurrent execution paths within a single process. The threads API is based on POSIX 1003.4 Draft 4. This component can map its calls directly to operating system threads, if operating system threads are available.

The DCE core services, and all depending applications, use threads. This all happens undercover. Customer applications may or may not use threads for their own purpose. However, application developers must know they *are* using threads anyway, through the DCE RPC runtime services.

If the operating system does not support threads in its kernel, the threads are running in (non-privileged) user mode. The kernel is then not aware of threads running in a process — it can only see (and dispatch) the process as a whole. The problem is that one thread could put the entire process into a wait state, thereby making all other threads also wait. Programmers have to be aware of this situation if they use threads. To avoid blocking the process with a thread, they should either use only calls from thread-safe libraries, use asynchronous I/O calls, or write their applications in a way that one server is only talking to one client at a time and vice versa.

2.2.2 DCE Remote Procedure Call

The DCE Remote Procedure Call (RPC) facility allows individual procedures in an application to run on a computer somewhere else in the network. DCE RPC extends the typical procedure call model by supporting direct calls to procedures on remote systems. RPC presentation services mask the differences between data representations on different machines and networking details to allow programs to work across heterogeneous systems.

DCE RPC has several advantages over traditional ONC RPC, the most important being its universal availability across many operating system platforms. It also features threads integration as well as naming and security integration. DCE RPC standards are adopted by X/Open, COSE, and ISIO.

DCE RPC provides programmers with several classes of functionality, like directory related RPCs, or security related RPCs, and powerful tools necessary to build client/server applications. Development tools consist of:

- Interface Definition Language (IDL) and related compiler idl. DCE IDL offers a more natural way of describing and defining interfaces of client/server code, and has more options compared to ONC RPC.
- uuidgen generates UUIDs (a 32-digit number) to uniquely identify resources, services and users in DCE independently from time and space.

- Runtime service implements the network protocol and communication between the client and server applications.

Using threads allows a client application to call several servers at once, for instance, for parallel calculation processes. A server, on the other hand, can serve multiple clients in parallel by using parallel running threads.

2.2.3 DCE Security Service

Distributed computing encourages a free flow of data between nodes, expanding the capabilities of interconnectivity and interoperability. Security breaches might come from any component of the distributed system. Security is one reason why customers are interested in DCE.

Security threats can be:

- Eavesdropping—Data can be read as it flows over the network.
- Masquerading—A system can pretend to be another system and thus gain unauthorized access to resources.
- Modification—Data can be modified as it flows over the network.
- Denial of service—Service can be denied from an unauthorized source.

These are just a few of the problems that can affect requirements, such as:

- Confidentiality—The information is disclosed only to authorized users.
- Integrity—The information is modified only by authorized users.
- Availability—The use of system, applications and services cannot be maliciously denied to authorized users.
- Accountability—Users are accountable for their security relevant actions.

For such reasons, security is a critical component in a distributed computing environment. A big concern is authentication of clients and servers. DCE solves this by using Kerberos. Some customers ask why they should pay thousands of dollars for DCE when they can get Kerberos free from MIT. Not only is DCE an authentication framework like Kerberos but also a complete security framework, with an architecture that enforces a discretionary security policy through the use of Access Control Lists (ACLs).

DCE embeds authentication and privacy into the RPC communication facility, thereby providing a powerful security framework to developers of DCE applications. If you were using only Kerberos without DCE, you would have to take care of the client/server communication and the encryption and decide whether you want authentication to happen just once or for each message. For more information on other security products and on encryption levels of DCE RPC, see Chapter 5, “Security in a Distributed Environment” on page 87.

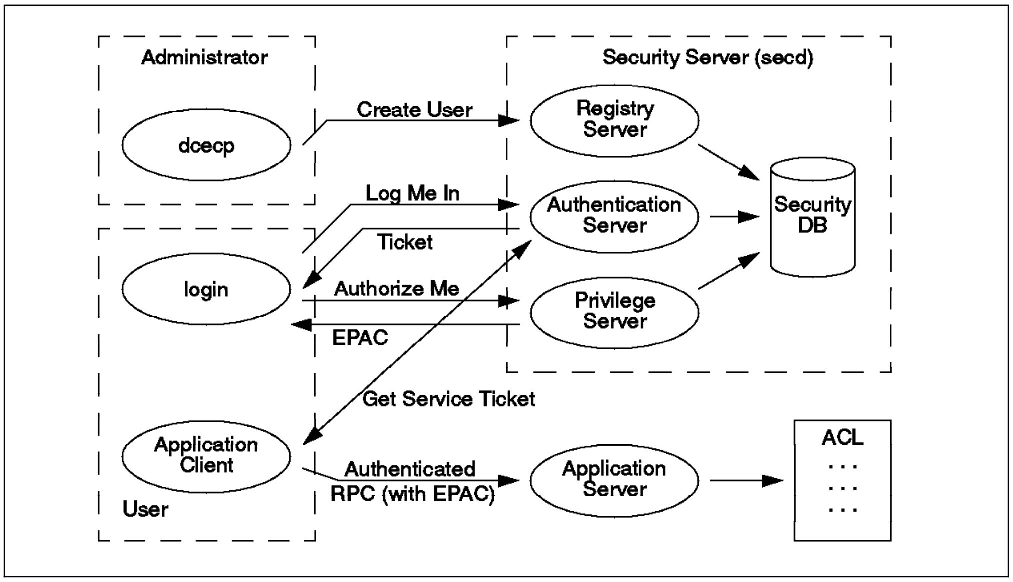


Figure 4. DCE Security Interactions

As depicted in Figure 4, interaction with the security service from user login to a service request works as follows:

1. The administrator uses the `rgy_edit` command or the `dcecp` command in OSF DCE 1.1 to administer user accounts in the Registry Service.
2. The identity of a DCE user or service is verified, or authenticated, by the Authentication Service.
3. The client gets a Ticket-Granting Ticket (TGT) that is encrypted with the user′s password. The ticket can only be decrypted if the user enters the correct password. The ticket contains a session key that the client needs to use for encryption/decryption in further communications with the security server.
4. The client, or more precisely the login facility, sends a request for authorization credentials, called the Privilege Attribute Certificate (PAC) or the Extended PAC (EPAC) in OSF DCE 1.1, to the Privilege Service.
5. The Privilege Service returns the PAC (or EPAC), which contains a list of groups, of which the principal is a member.
6. If the client wants to contact a remote RPC server, it needs a service ticket from the security server. This ticket contains the requested server′s session key that is used to encrypt/decrypt messages between the client and the server.

   The RPC request that is sent to the server contains the ticket with the PAC. It is then basically up to the application server to decide whether it lets the user do what they want based upon the info in the PAC. If the application has implemented an ACL manager, it can send the request to the ACL manager for authorization against the ACL defined on the resource that the client wants to access. The advantage of implementing an ACL manager is that permissions can be edited and accessed with standard methods (`acl_edit` command, RDACLIF API).

The security service can be replicated. Each security slave server holds a complete registry database. Updates are only possible to the master security server, which immediately tries to update all its slave servers. A cell only works if the security service can be reached to get tickets. The replica servers are able to issue tickets. So, creating one or more replica servers increases the availability of your DCE cell security service and also provides load balancing. However, too many security servers would create network overhead, increase the risk of being cracked, and are expensive because they need a license for each server instance. Clients randomly choose a security server by asking the cell directory service (CDS) for binding information. Therefore, installing a security server in each location does not help unless you make some static definitions on each client machine to bypass CDS and bind to a specific security server. Careful planning is necessary to balance all the different requirements and aspects.

2.2.4 DCE Directory Service

The DCE Naming Service provides a naming model throughout the distributed environment. This model allows users to 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 in a network. Further, users can continue referring to a resource by the same name even when a characteristic of the resource changes, such as its network address.

The global distributed computing environment is composed of administratively independent cells. The name space is hierarchically organized and forms a tree, with containers (directories) and leaf objects. The root directory, /..., is global and contains all cell names, which build the root directories for each cell. The subtrees underneath each /.../<cellname> directory are under the management domain of their respective cell. Users within a cell can use the short form /.: to address their local root directory. CDS then replaces the name /.: with the appropriate global name /.../<cellname>. The leaf objects can be any resource, as mentioned above. The main purpose, however, is to store and retrieve binding information for DCE RPC servers that can be used by DCE clients to find and bind to a server.

The naming system consists of the following components:

- Cell Directory Service (CDS)
- Global Directory Service (GDS) X.500 (separate product)
- Global Directory Agent (GDA)
- Application Programming Interface

The local naming system is provided by the CDS and can be distributed and replicated. It is integrated with a global naming system, X.500 or DNS (Domain Name System), via a Global Directory Agent. Communication between cells is done via the Global Directory Agent (GDA), where the global name space is the naming bridge. The global name space here can be X.500 or DNS.

Objects from the global name space are available from within a cell via the Global Directory Agent (GDA) function, which translates CDS internal protocol to OSI DAP (Directory Access) protocol. The GDA supports worldwide DCE access via DNS, the TCP/IP Domain Name Service or CCITT X.500, which is provided by GDS. The GDA is the CDS gateway to GDS. Both CDS and GDS offer the X/Open Directory Service (XDS) API as their programming interface.

The following example shows a CDS object (AIXDB1) with two levels of subdirectories (subsys/database) in a cell austin.ibm.com. The first line shows the global name by which the object can be found from any other (connected)

DCE cell in the world, and the second line shows the cell-relative name of the same object as it can be used from users logged in to cell austin.ibm.com:

```
/.../austin.ibm.com/subsys/database/AIXDB1
/.:/subsys/database/AIXDB1
```

The CDS can be replicated. Each CDS server holds its own database, a clearinghouse. Replication is performed on the basis of directories. Each directory may be replicated in as many CDS servers as you wish. All replicas of a directory are called the *replica set*. When defining the replica set, one instance has to be declared the master (read/write copy), the others are read-only. When a master is changed, it updates the read-onlys of its replica set. Installing a new CDS server creates an empty clearinghouse that must be explicitly populated with the replicas it is to contain. So, not all directories need to be in all clearinghouses, and the master replicas can be in any CDS server. This means that there is no master CDS server nor read-only CDS servers; a CDS server can contain master replicas of some directories and read-only replicas of other directories. CDS is a distributed, replicated service.

A cell usually only works if the CDS can be reached to get binding information (server addresses). This can be achieved from read-only replicas. DCE servers need to be able to export their binding information, which requires write access to the master replicas. So, if there are unreliable network connections over a WAN, it may be a good idea to install a CDS server in a remote location and define the master replicas of important location-specific directories there. DCE service binding information can also be statically defined in CDS, but then it is there even if the service is not available, and clients trying to access the service experience time-outs. Additional CDS servers increase availability and perform load balancing, but also create network and administration overhead and require a license for each server instance.

2.2.5 DCE Distributed Time Service

The Distributed Time Service (DTS) provides precise, fault-tolerant clock synchronization on the computers participating in a Distributed Computing Environment over LANs and WANs. The synchronized clocks enable DCE applications to determine event sequencing, duration, and scheduling. The core services, especially the ticket granting service, heavily rely on synchronized clocks. Note that installing DTS is not a requirement; the clocks could be synchronized by other time services. However, the use of DTS is recommended because it uses security service and adjusts time smoothly rather than correcting system clocks all at once or even backwards. The DTS clerks obtain time information from at least three DTS servers in a LAN and adjust their time. If they do not receive the required number of time values in their LAN, they contact global DTS servers. The synchronization process of DTS is able to detect certain fault situations, providing some fault-tolerance. If, for example, a time server's clock drifts away too fast or changes it's time at once, it will be automatically taken off the synchronization process.

DTS is based on Coordinated Universal Time (CUT or UTC), an international time standard. Different types of time servers provide for different transmission delays between LANs and WANs which would influence correct time calculation:

- Local DTS Servers: Maintain synchronization within a LAN and synchronize their own clocks using the responses of all other DTS servers in the LAN. If they do not get responses from at least two other DTS servers in their own LAN, they have to contact global DTS servers.

- Global DTS Servers: Usually at least one per LAN. They advertise themselves in the CDS so they can be contacted by other DTS servers or even clerks if they do not have the required number of DTS servers in their own LAN. To adjust their own clocks, they act like local DTS servers. If they get their time from an external time provider, they do not adjust their clock with values obtained from other DTS servers.
- Courier DTS Servers: Usually one per LAN. Any local or global DTS server can have a courier role. What is special about this role is that a courier must contact one global DTS server, even if it gets enough time values from DTS servers in its own LAN. By doing so, couriers maintain synchronization between multiple LANs.

A complete set of DCE DTS APIs is offered as well as a Time Provider Interface (TPI) that allows a time provider process to pass its UTC time values to a DTS server. Many standards bodies disseminate UTC by radio, telephone, and satellite. TPI also permits other distributed time services, such as the Network Time Protocol (NTP), to work with DCE.

2.2.6 Distributed File System

DCE DFS is a distributed file system that allows users to share files stored in a network of computers as easily as files stored on a local machine/workstation. The DCE Distributed File System uses the client/server model common to other distributed file systems. The file system gives users a uniform name space, file location transparency, and high availability. Reliability is enhanced with a log-based physical file system that allows quick recovery after server failures. Files and directories can be replicated to multiple machines to provide reliable file access and availability. Security is provided by a secure RPC service and Access Control Lists, which conform to POSIX 1003.6. DFS implements a superset of that POSIX ACL Draft.

As shown in Figure 3 on page 11, DFS is an Extended Service and is built on the DCE core services: Security, CDS, and DTS. When accessing remote data, DFS uses DCE Remote Procedure Calls (RPCs) to communicate between participating systems in exchanging authorization requests, access requests, file and directory data, and synchronization information. It uses the DCE Naming Services to resolve global names and the DCE Security Service to authenticate users and services. It depends on the DCE Time Service to keep the clocks in synchronization.

The DCE LFS is a log-based file system that is integrated into the kernel. Also, it is based on aggregates that are equivalent to standard UNIX disk partitions or AIX logical volumes. Aggregates are logically composed of multiple filesets, which are mountable subtrees. Filesets share the disk blocks within an aggregate. Filesets can be administered and referenced individually. Quotas can be set on a per-fileset basis. Filesets are the units that provide support for administrative functions, such as replication, cloning, reconfiguration (move filesets for load balancing), and backup, needed in a distributed environment. The cloning function provides copy-on-write semantics so that double disk space is not needed when a fileset is cloned. Cloning also allows the above-mentioned functions to be performed while the filesets are online, with minimal down time for users of the filesets.

Directories and files can be accessed from users anywhere on the network, using the same name since all DCE resources are part of a global namespace.

High performance is achieved through caching on the client side to reduce access time and network traffic.

DFS has many advantages over NFS:

- Stateful implementation allows for caching on the client side
- Provides single-site read/write semantics
- Fileset replication
- Security (Authentication and ACLs)
- Cloning
- Backup servers

DFS files can be exported to NFS so that NFS clients can access them as unauthenticated users. The new NFS/DFS Authenticating Gateway product provides a mapping of NFS users into authenticated DFS users. To achieve this, NFS users use the `dfsiauth` command to perform a DCE login to set up credentials for a certain combination of userID/nodeID, which will be revoked when the ticket expires.

2.3 Organization of DCE Networks

In DCE, a cell represents the smallest unit of resources, such as systems, users, services, and nodes that work, and are administered, together.

A minimal cell must include threads, the RPC communication layer, and at least one instance of all the core services:

- Cell Directory Server
- Security Server
- At least three Distributed Time Servers (optional, but recommended)

Cells can be defined and configured in different ways, depending on the user, administration, or company requirements. For example, a small company that offers only one kind of service can be set up as a single cell. Another example might be the faculty departments at the University of Texas. They can have their own manageable cells and use inter-cell communication for common services or data.

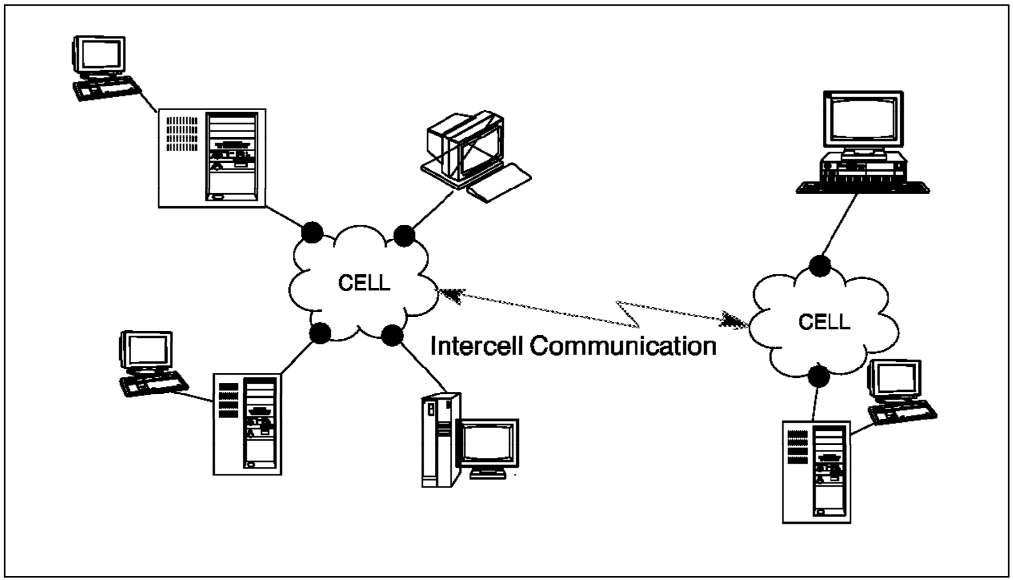


Figure 5. DCE Multi-Cell Environment

Inter-cell communication is provided through GDA. The DCE architecture supports different types of network protocol families. The current OSF DCE reference implementation runs over the Internet Protocol (IP) family, using either UDP (User Datagram Protocol) or TCP (Transmission Control Protocol) as transport layers.

The home cell for a principal shows the cell where the information about the principal is stored. More generally speaking, a cell represents the collection of resources that use a common naming and security policy.

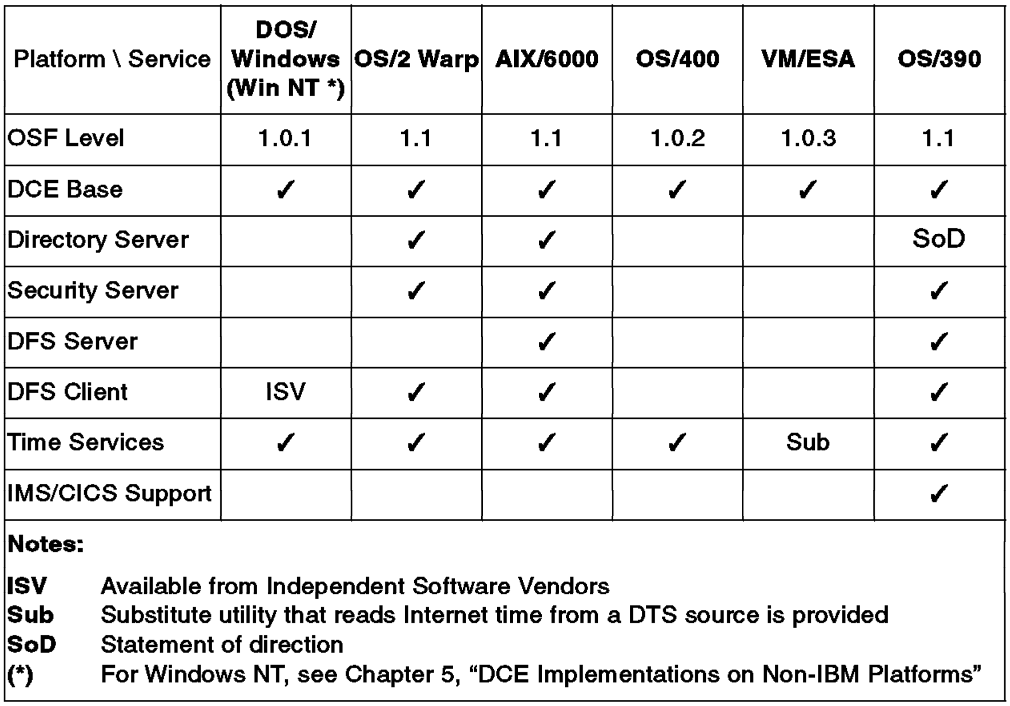
Chapter 3. DCE Implementations on IBM Platforms

The purpose of this chapter is to help customers, system engineers, and marketing representatives understand how DSS or DCE is implemented on the different IBM platforms and what DCE functions are available. In order to understand the DCE implementation on platforms with which the reader might be unfamiliar, we start off with an explanation of the operating system environment that enables DCE to run. For instance, we discuss what process or thread is on each platform and how the file system looks.

Each platform is described in the same way:

- Operating system environment - explains some OS basics
- DCE implementation specifics - explains how DCE is implemented
- User environment - explains how user logins are integrated
- Administration tools - explains what commands and tools are available
- Application development - explains what development tools are available
- DCE products and packaging - lists order numbers and prerequisites

Table 1 gives a cross-platform summary of the current DCE implementation status.



Platform \ Service	DOS/ Windows (Win NT *)	OS/2 Warp	AIX/6000	OS/400	VM/ESA	OS/390
OSF Level	1.0.1	1.1	1.1	1.0.2	1.0.3	1.1
DCE Base	✓	✓	✓	✓	✓	✓
Directory Server		✓	✓			SoD
Security Server		✓	✓			✓
DFS Server			✓			✓
DFS Client	ISV	✓	✓			✓
Time Services	✓	✓	✓	✓	Sub	✓
IMS/CICS Support						✓

Notes:

ISV Available from Independent Software Vendors
Sub Substitute utility that reads Internet time from a DTS source is provided
SoD Statement of direction
(*) For Windows NT, see Chapter 5, “DCE Implementations on Non-IBM Platforms”

Table 1. IBM DCE Cross-Platform Matrix 11/96

The last section contains a more comprehensive function/product overview of the DCE implementations on each platform. See 3.7, “IBM DCE Cross Platform Matrix 11/96” on page 78 for the current cross-platform matrix.

3.1 OS/390 Implementation of DCE

IBM's OS/390 operating system is XPG4.2 (X/Open Portability Guide) UNIX 95 branded. Having achieved XPG4.2 UNIX 95 certification, OS/390 offers full UNIX capabilities built directly into the operating system. OS/390 can provide UNIX facilities in tandem with OS/390 classic strengths like availability, reliability, security, and scalability.

One of the many new features in OS/390 is DCE support based on OSF DCE Version 1.1. This includes DCE clients and servers, DFS client and server, and DCE integration with core components in OS/390 like RACF, CICS, and IMS.

IBM has made DCE services available to both the traditional mainframe and the new UNIX environment, and this will allow developers, users, and administrators to use DCE services within both environments.

IBM offers all the OSF DCE Version 1.1 services in OS/390, except the DCE Directory and the Global Directory servers, and X.500 runtime support. DCE Directory server for OS/390 is in development and is expected for release in 1997.

This section first describes the UNIX environment in OS/390. If you are already familiar with the UNIX environment in OS/390, then skip to 3.1.2, "DCE Implementation Specifics on OS/390" on page 28, which describes the DCE services and how the DCE services are integrated with other components in OS/390.

3.1.1 OS/390 UNIX Environment

The purpose of this section is to describe the X/Open XPG4.2 UNIX 95 standard and how it is implemented on the OS/390 platform because this is the basis for OSF DCE services on OS/390.

3.1.1.1 X/Open XPG4.2 UNIX 95 Overview

In September 1993, the computer systems industry, in the form of a group of 75 leading system and software vendors, agreed to ask X/Open to standardize the specifications of the UNIX operating system. In a parallel move, Novell, Inc. announced that it would transfer the UNIX trade mark, which it owned, to X/Open Company Ltd.

A few weeks and two announcements later, the computer industry put aside over 25 years of policy and technical differences to concentrate on delivering a single specification for UNIX to the market with the aim of creating a consistent volume market for UNIX systems. All major vendors, including those that would continue to develop and maintain their own implementations of their operating systems, agreed that they would ensure that their operating system products would meet a single comprehensive standard specification for APIs (application programming interfaces).

This industry initiative worked through the proven processes of X/Open to develop a set of specifications by broad industry consensus, supported by a testing and branding program, that would deliver commonality across multiple implementations of UNIX systems. The early specification (known then as Spec 1170) took into account the interfaces utilized by the most widely used UNIX applications to ensure legacy-application compatibility. This single UNIX specification now defines which products may be called “UNIX” and is backed by the X/Open branding program.

To support the branding program, X/Open has developed a comprehensive set of software test tools. These tests are used by the vendors to ensure that their products comply with the specification.

OS/390 has achieved XPG4.2 UNIX 95 certification by X/Open Company Ltd. The UNIX 95 specification is the latest specification from X/Open Company Ltd. at the time this book was written.

The result is that OS/390 now offers UNIX services, based on open standards, and at the same time provides the classic strengths of the MVS environment, like availability, reliability, security, and scalability.

The X/Open Company Ltd. has defined a set of UNIX standards with XPG4.2 UNIX 95, which include POSIX standards. The XPG4.2 UNIX specifications have been integrated into OS/390, and OS/390 now supports more than 1100 UNIX functions. See also figure Figure 6 on page 24.

The XPG4.2 support in OS/390 consists of:

- **Application Programming Interface (API)** - XPG4.2 defines a set of system calls that are currently defined in terms of C language bindings. These system calls are partially accessible from other High Level Languages (HLL) in OS/390 through the Assembler callable service interface.
- **UNIX Shell Interface** - OS/390 provides an XPG4.2 compliant shell interface, that has the look and feel of a UNIX system. Standard UNIX utilities are also provided. The shell interface and applications can invoke shell scripts that are like Time Sharing Option (TSO) clients. This interface and utilities are provided for “portability” of UNIX skills.
- **Customer Applications** - Both the application programming interface and the UNIX shell interface are compatible to the interface found on a typical UNIX system from other vendors who support XPG4.2 specifications. Using this interface, the customer can easily write or port UNIX applications to OS/390. Portability requires moving the UNIX source code, recompiling the application, and testing on OS/390.
- **Hardware Management** - UNIX systems provide kernel code which manages the hardware and controls the resources used by UNIX applications, called processes. OS/390 utilizes the hardware and task control mechanisms that existed in the MVS system. Thus, all the reliability and error recovery built into the supervisor apply to UNIX applications on OS/390.

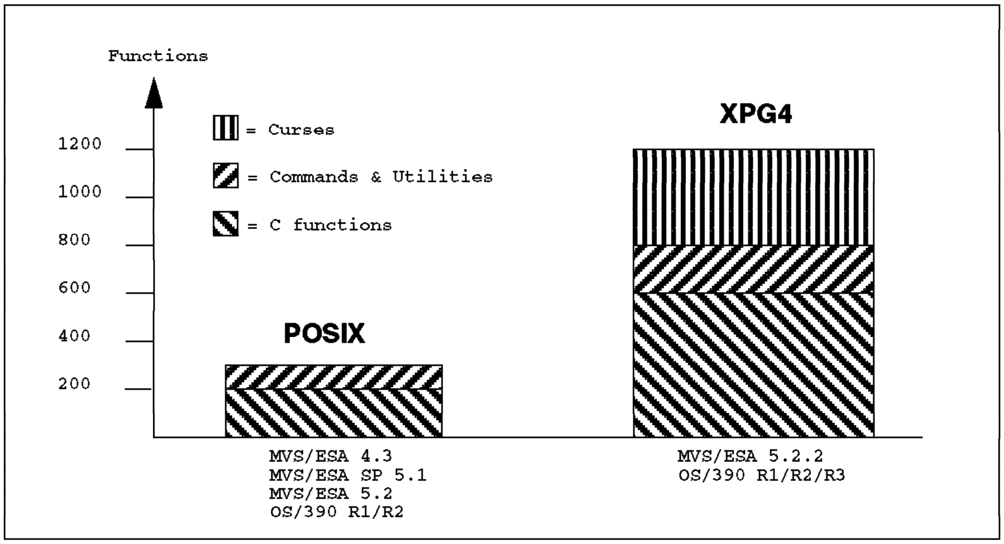


Figure 6. POSIX and UNIX with OpenEdition MVS and OS/390

3.1.1.2 OS/390 UNIX System Environment

OS/390 supports the concurrent running of UNIX applications within the same processor or sysplex environment as other heritage applications. UNIX applications are loaded into the address spaces and given use of the hardware by the OS/390 supervisor.

The OS/390 integrated communication server can be used to interface to UNIX clients in the network. In addition, DCE applications using the DCE support can also use the integrated communication adapter.

Data management for the UNIX file system and tapes is provided by the Data Facility Product (DFP) element of OS/390.

Security support for UNIX and DCE applications is provided by RACF or an equivalent security package. See also Figure 7 on page 25.

Because the UNIX services defined by XPG4.2 UNIX 95 specifications are integrated into the OS/390, they are available to OS/390 applications using traditional MVS functions that have been carried forward into OS/390. While some applications will only use XPG4.2 functions to allow portability across XPG4.2-compliant platforms, others may use additional functions in OS/390 to optimize the application. One example of an application that is mixed is one which accesses both MVS data sets and the XPG4.2 UNIX file system in OS/390. Such applications can provide a bridging function between the two operating system environments, UNIX and MVS, on a single operating platform.

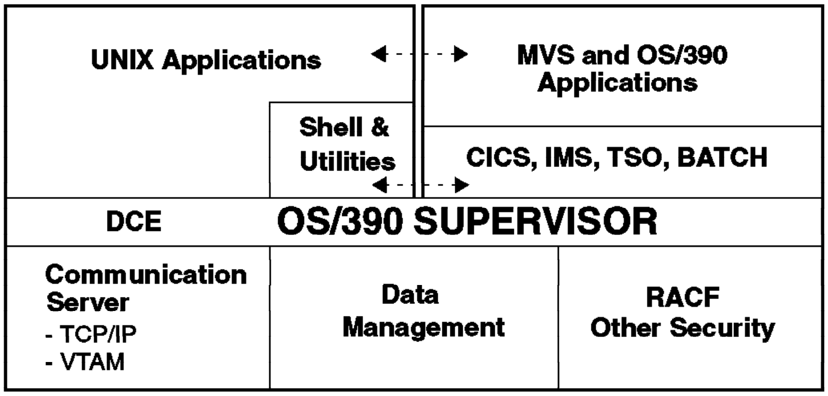


Figure 7. OS/390 UNIX System Environment

3.1.1.3 The UNIX File System in OS/390

One of the most fundamental differences between XPG4.2 and MVS is the XPG4.2 file system. By definition, an XPG4.2 compliant file system must be hierarchical and must be byte-oriented. Such a file system has been provided in OS/390 as a capability of the DFP element. See also Figure 8 on page 26.

The hierarchical file system:

- The XPG4.2 HierarchicalFile System (HFS) is very similar to hierarchical file systems on DOS or OS/2. Investment in data is protected because files stored on the file system are portable to a wide range of platforms that support UNIX data.
- Finding a file in the XPG4.2 file system is done by searching a directory of a series of directories. There is no concept of an MVS catalog that points directly to a file.
- A pathname identifies a file and consists of directory names and a filename.
- A fully qualified filename, which consists of the names of each directory in the path down to a file plus the file name itself, can be up to 1024 bytes long.
- The HFS file system allows for file names in mixed case.

OS/390 integration:

- The integration of HFS with existing OS/390 file-system-management services provides automated file-system-management capabilities that may not be available on other XPG4.2 platforms. This allows file owners to gain productivity by spending less time on tasks such as backups and restores of entire file systems.
- The file system has been designed to extend into the DCE Distributed File System (DFS).
- HFS is managed by the DFP element of OS/390.
- HFS security is integrated with the security of Resource Control Access Facility (RACF) on OS/390 through the System Authorization Facility (SAF) interface.
- Special files called pipes and FIFO (First-In First-Out) files allow for exchange of information between processes or threads.

- Terminals are accessed like files.
- Character-special files allow the file system to use communication device drivers. This support provides interactive terminal I/O in an XPG4.2 environment.

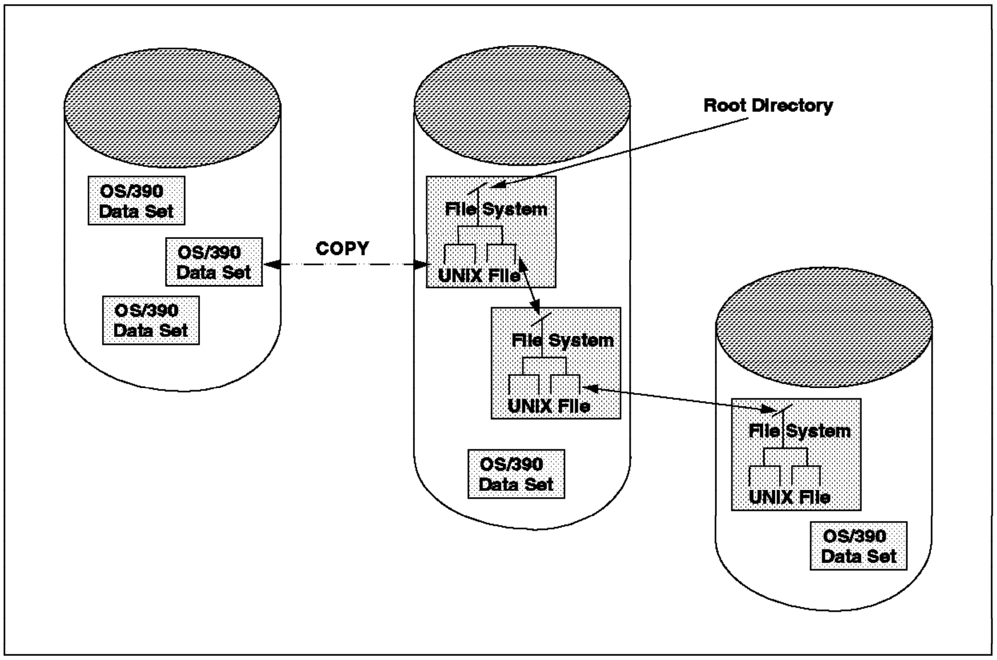


Figure 8. Hierarchical File System in OS/390

3.1.1.4 Shell and Utilities in OS/390

An XPG4.2 shell environment is like a TSO environment in that it provides an interactive environment for developing and running applications. It provides for administrator functions as well as user functions.

A user requests the XPG4.2 shell environment from a TSO session. Once in the XPG4.2 shell environment, you can toggle back to the TSO environment to perform TSO functions and then return to the XPG4.2 shell environment. Since the XPG4.2 shell environment is accessed from a TSO session, you can access the XPG4.2 shell environment from a 3270, SNA, or TCP/IP 3270 emulator session.

With OS/390, the XPG4.2 shell can also be accessed directly by using UNIX commands from other systems, such as `rlogin` or `telnet`. The TSO environment is not available with this direct login; however, a TSO command may be entered at the shell prompt.

IBM has licensed Mortice Kern Systems' InterOpen XPG4.2 shell and utilities source code to bring OS/390 into conformance with XPG4.2 standards. The shell is like the TSO Terminal Monitor Program (TMP). It owns the terminal, interprets the command string, and invokes the requested commands. A set of built-in commands is provided to the user.

UNIX applications can be run on OS/390, which uses the X-Windows or Motif support supplied by the TCP/IP in the OS/390 communication server. GUI interface is supported using X-Windows and Motif.

UNIX applications can be run on OS/390 which support 3270 real or emulated terminals.

3.1.1.5 POSIX Threads in OS/390

With threads support, OS/390 application developers are provided with a general set of services to develop multitasking server applications that require concurrent execution streams. Applications using thread services may be designed to use fewer system resources and less storage than similar programs using fork. For example IMS TM application developers can use threads. Additionally, the OS/390 DCE support uses threads.

Threads are defined as multiple dispatchable units of work within a process. A POSIX threads is loosely equivalent to an MVS task. The following figure illustrates one process with an initial thread and four sub threads. The process is just one of many processes in the OS/390 address space.

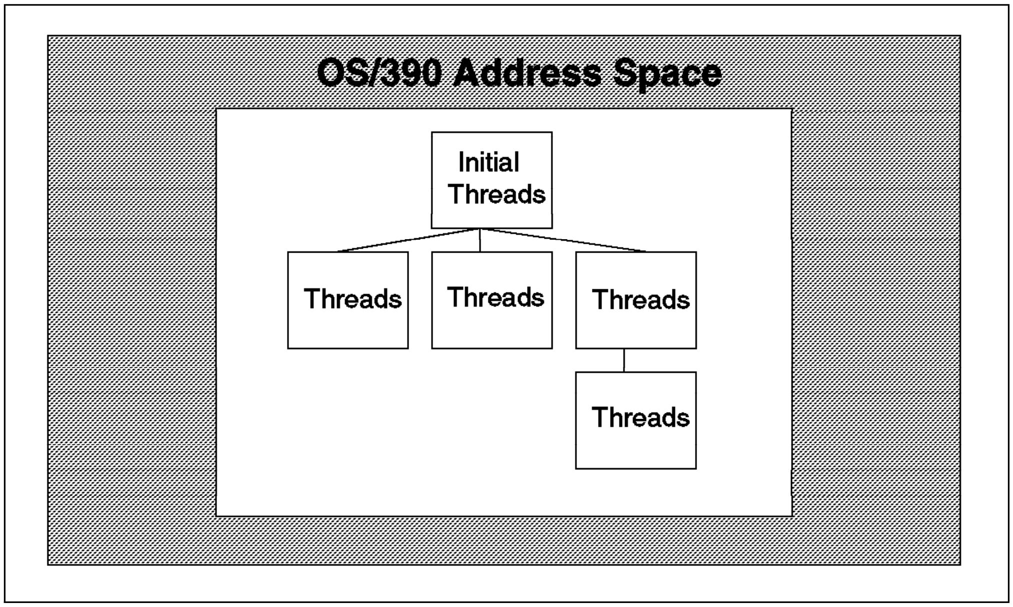


Figure 9. Threads Support in OS/390

3.1.1.6 Socket Applications

Socket is the application interface UNIX provides for applications that communicate via TCP/IP. OS/390 provides socket support for standard UNIX applications. The OS/390 socket application environment is enhanced to support socket application transport over an SNA network via the VTAM AnyNet/MVS feature. With this support, integrated sockets can run over SNA and TCP/IP.

For DCE applications, the VTAM AnyNet/MVS feature supports a DCE application transport over SNA networks. This support permits OS/390 DCE applications to run over either SNA or TCP/IP networks. See also Figure 10 on page 28.

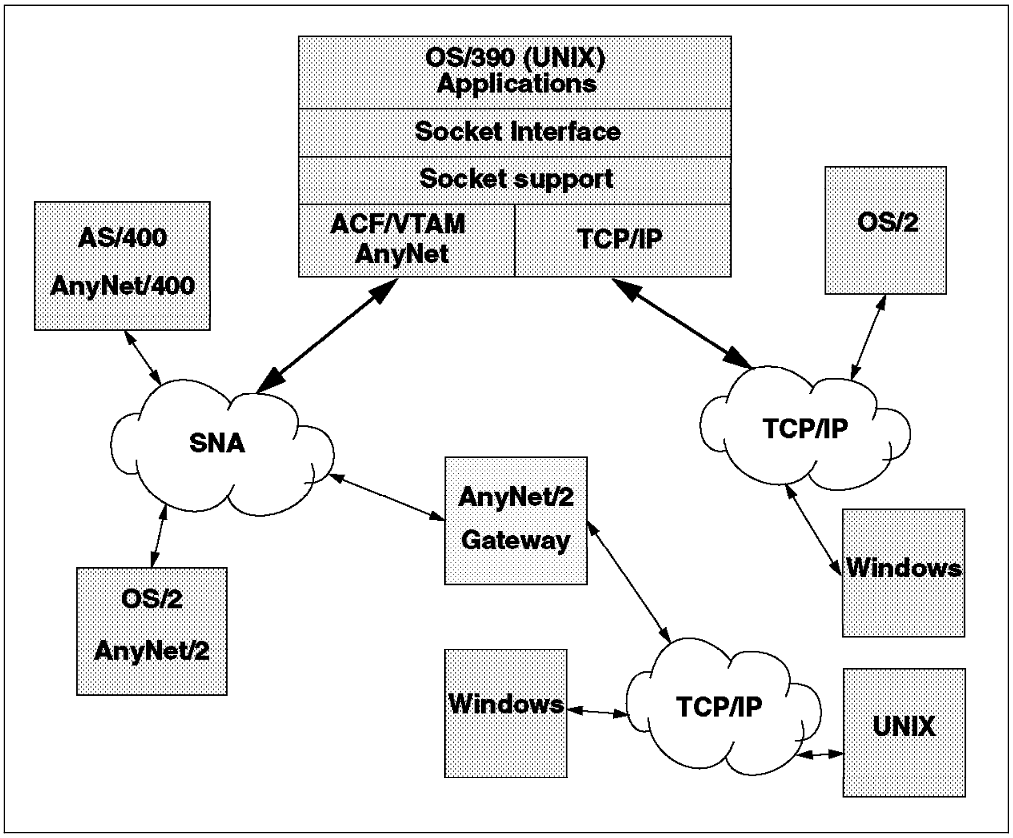


Figure 10. Networking Choices in OS/390

3.1.2 DCE Implementation Specifics on OS/390

This section describes what DCE components are available on OS/390 and how they are implemented in the operating system.

The following DCE components are available for OS/390:

- OS/390 DCE Base Services, based on OSF DCE Version 1.1. This includes the DCE runtime, security, directory, time services plus application development tools, headers and libraries. These services are included in the OS/390 operating system. For more information, see 3.1.2.1, “OS/390 DCE Base Services” on page 29.
- OS/390 Security Server optional feature. This service includes RACF Version 2 Release 2 and OSF DCE Security server, based on OSF DCE Version 1.1. These services are added on to the OS/390 operating system. For more information, see 3.1.2.2, “OS/390 DCE Security Server” on page 29.
- OS/390 DCE User Data Privacy (DES and CDMF) or OS/390 DCE User Data Privacy (CDMF). These services are optional features to the base OS/390 system. For more information, see 3.1.2.4, “OS/390 Encryption Services” on page 30.
- OS/390 DCE Distributed File Service (DFS), based on OSF DCE Version 1.1. OS/390 DFS includes both the DFS server and client. DFS services are included in the OS/390 operating system. For more information, see 3.1.2.5, “OS/390 Distributed File Service” on page 31.

3.1.2.1 OS/390 DCE Base Services

The DCE base services are one part of OS/390 DCE Base Services. The following daemons perform the core functionality:

- A DCED daemon. This daemon takes care of RPC client and server programs and the RPC end-point mapper. DCED also handles security validation services.
- A CDS advertiser daemon that handles access to CDS servers and starts the CDS clerks.
- A CDS clerk daemon. The CDS clerk handles requests from the client application to a server and caches the results.
- A Time client daemon. The Time client daemon is combined with the Time server daemon, if this is configured and started.

The DCE daemons run as individual daemons in the DCE kernel address space. They are started and stopped through the control task. The control task also detects if the daemons have prematurely stopped and tries to restart them.

The OS/390 DCE Base Services also includes a DCE Time Service Time Provider. It maintains a separate DCE software clock. This will allow DCE Time Service to relay on OS/390 hardware clock. An OS/390 host does not allow an external agency to tamper with the hardware clock. To prevent the two clocks from shifting apart, it is recommended to run the DCE Time Server and configure the time provider as a null-time provider for the DCE cell.

3.1.2.2 OS/390 DCE Security Server

The OS/390 DCE Security Server provides a fully functional OSF DCE Version 1.1 Security Server that runs on OS/390. You can configure your DCE cell with a security server on the mainframe. The OS/390 DCE Security Server provides the following benefits:

- You can keep servers and data on your OS/390 systems safe from accidental or malicious loss and secure from outside attack (disclosure or corruption).
- System/390 is scalable and many OS/390 systems are large. You can build DCE cells that are able to handle large numbers of accounts.
- OS/390 is more reliable than most other operating system platforms currently offering DCE. Therefore, a security server on OS/390 is more reliable and available than on most other DCE systems, giving you the reliability you need to run mission-critical applications in a DCE environment.
- OS/390 DCE provides interoperability between Resource Access Control Facility (RACF) on OS/390 and OS/390 DCE. This security interoperability allows a DCE client to access a DCE-enabled server on an OS/390 DCE system and allows the DCE server to acquire corresponding local security credentials for a DCE client to access OS/390 resources. The interoperability function allows:
  - Appropriately authorized DCE servers to acquire corresponding OS/390 security credentials for the DCE client and use the DCE client′s corresponding OS/390 user ID for access to RACF-authorized resources.
  - An OS/390 user to be transparently logged in to DCE when necessary, without prompting for a DCE user ID or password. This ability is called single sign-on. With this feature, an MVS user authenticates to OS/390 and can start a DCE program without reauthenticating to DCE.

3.1.2.3 RACF Interoperability and Single Sign-On

OS/390 DCE also provides interoperability between RACF on OS/390 and OS/390 DCE. This security interoperability allows a DCE client to access a DCE-enabled server on an OS/390 system and allows the DCE server to acquire corresponding local security credentials for a DCE client to access OS/390 resources. The interoperability function allows:

- Appropriately authorized DCE servers to acquire corresponding OS/390 security credentials for the DCE client and to use the DCE client′s corresponding OS/390 user ID for access to RACF-authorized resources.
- An OS/390 user to be transparently logged into DCE when necessary, without prompting for a DCE user ID or password. With this single sign-on feature, an OS/390 user authenticates to OS/390 and can start a DCE program or access any other DCE resources without reauthenticating to DCE.

OS/390 DCE also provides administration utilities to implement the interoperability. These incorporate the information into RACF that associates an OS/390 user ID with a DCE principal′s identifying information and the DCE principal′s UUID with the corresponding OS/390 user ID. This is called cross-linking information, and it is what allows interoperability and single sign-on to work.

The cross-linking information must be set up before interoperability functions can be used. To do this, DCE provides two utilities, mvsimpt and mvsexpt, for creating the initial cross-linking between the two registries. This cross-linking can be done from either the RACF database or the DCE registry, but mvsimpt and mvsexpt must be started from the MVS system where the RACF database whose users are to be cross-linked resides. The individual user can use the storepw utility program to store his DCE password in the RACF DCE segment. This enables the undercover DCE login.

OS/390 DCE also provides application programming interfaces (APIs) to the SAF so that you can write your own server programs to take advantage of RACF interoperability.

3.1.2.4 OS/390 Encryption Services

OS/390 delivers optional features that provide DCE encryption services to DCE. DCE internally use DES encryption, but if the encryption facility is going to be used for applications, using the privacy option for authenticated RPC, then one of the following options must be installed on OS/390.

- OS/390 DCE User Data Privacy (DES and CDMF). This option is the default option for USA and Canada. This option requires special export license for countries outside USA and Canada.
- OS/390 DCE User Data Privacy (CDMF). This option is for countries outside USA and Canada.

The export restrictions for the DES encryption technology may change during 1996/1997.

3.1.2.5 OS/390 Distributed File Service

OS/390 provides DCE Distributed File Service (DFS) client and server functionality.

For end users, DFS provides the following benefits:

- Access to DFS data on OS/390 without regard to physical location from anywhere in the distributed computing environment.
- Consistency among distributed files.
- Improved response time as a result of local caching.
- Improved availability of data.
- Improved security.
- The DFS server also supports export of MVS Record Data. This includes datasets of the following types: Sequential (SDS), Partitioned (PDS & PDSE) and VSAM (KSDS, ESDS, RRDS).

For DFS administrators, DFS provides:

- File sharing between the OS/390 Hierarchical File System (HFS) and the DCE Local File System (DCE LFS).
- Directory and Security Services integrated with OS/390 DCE.
- Improved availability of data.
- The ability to perform routine server maintenance, such as load balancing and data backups, while the server data is still in use by end users.
- The ability to use S/390 system management software and high-speed hardware devices.
- DCE Access Control List (ACL) support on LFS.

DFS server machines run processes that provide services such as making data available and, on an administrative level, monitoring and controlling other processes. Server machines are defined by the processes/daemons they run. Among the processes DFS server machines run are these:

- The File Exporter (DFSKERN) fills requests for data from client machines anywhere in the network.
- DFSXPORT determines the data to be made available to DFS clients anywhere in the network.
- The Basic OverSeer Server (BOSERVER) monitors the DFS processes that run on a server and restarts them when needed.
- The Backup Tape Coordinator (BUTCnn) controls a backup tape drive and accepts service requests for backup.
- The Replication Server (RPSERVER) copies read-only replicas of read/write data onto other file server machines.
- The System Control Server (UPSERVER) updates other server machines with identical versions of system configuration files.
- The System Control Client (UPCLIENT) retrieves system configuration files from the system control server.
- The Fileset Server (FTSERVER) provides an interface to the DFS commands and components used to manipulate filesets.

- The Backup Database Server (BAKSERVER) houses the master and replica versions of the Backup Database, where information used to back up and restore system and user files resides.
- The Fileset Location Server (FLSERVER) houses the master and replica versions of the Fileset Location Database (FLDB), where information about the location of systems and user files is maintained.

DFS client machines, generally workstations, provide users with computational power, access to DFS files, and other general-purpose tools. The Cache Manager requests data for users from the processes running on file server machines. A client Cache Manager on a non-OS/390 DFS system can work in conjunction with OS/390 DFS to improve file access performance.

When the Cache Manager receives a packet of data for an HFS or DCE LFS file from a file server machine, it stores copies of the data in a local storage area called the cache. The cache is an area reserved for data storage on the local disk or in memory on the client machine. The Cache Manager uses the local copy of the cached file; it does not continue to send network requests to file server machines for data it has stored locally. Access to locally stored data is much quicker than access to data stored across the network.

As you save the changes you make to data, the Cache Manager periodically sends the changed data back to the appropriate file server machine, where your changed version replaces the data stored on the server. When the central version of the file stored on the file server machine changes, DFS advises all other Cache Managers with copies of the file that their versions are no longer current. When other users access the file, their Cache Managers use the newer version of the data.

3.1.3 User Environment

An OS/390 user can get access to DCE from either the TSO or the UNIX environments in OS/390. Access to the UNIX environment is provided either directly by using a Telnet session or through TSO.

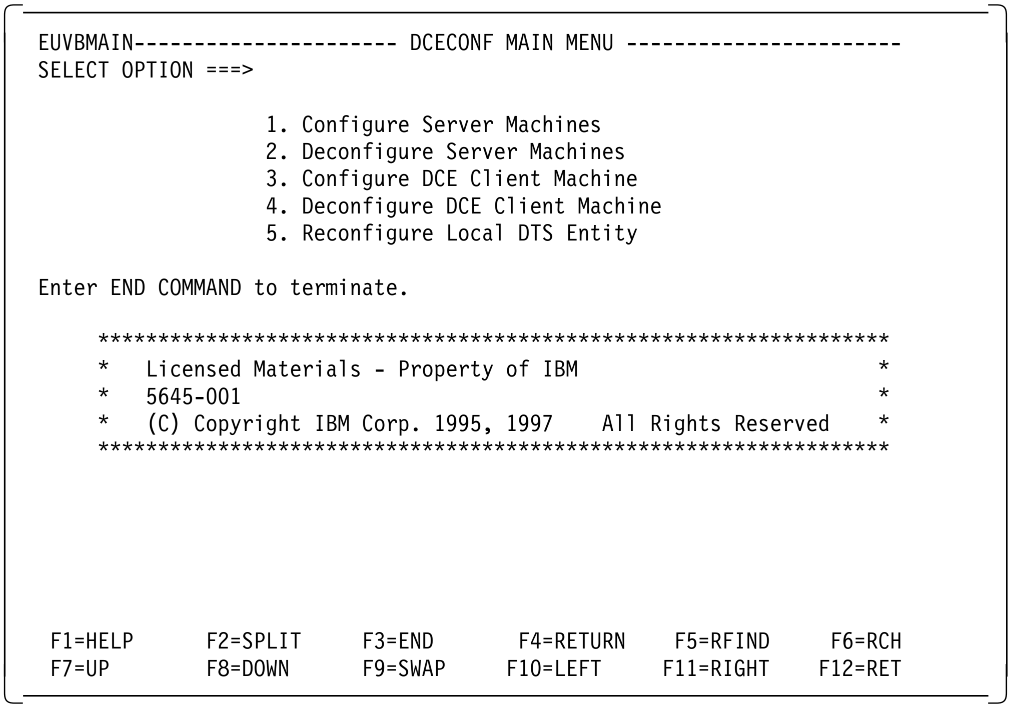
Since the UNIX environment in OS/390 provides all the native DCE tools and utility programs, the UNIX environment is the best place to function for DCE application developers and DCE administrators.

Basically, there is not very much to do for the user with respect to DCE. And since IBM has provided single sign-on between RACF and DCE, the user may not even need to use the `dce_login` program to log in to DCE.

3.1.4 Administration Tools

The `DCECONF` and the `DFSCONF` programs are used to configure the DCE services. The commands are available from TSO.

Figure 11 on page 33 is an example of the `DCECONF` program to configure DCE.



```
EUVBMAIN---------------------- DCECONF MAIN MENU -----------------------
SELECT OPTION ===>

                   1. Configure Server Machines
                   2. Deconfigure Server Machines
                   3. Configure DCE Client Machine
                   4. Deconfigure DCE Client Machine
                   5. Reconfigure Local DTS Entity

Enter END COMMAND to terminate.

     ******************************************************************
     *   Licensed Materials - Property of IBM                         *
     *   5645-001                                                     *
     *   (C) Copyright IBM Corp. 1995, 1997    All Rights Reserved    *
     ******************************************************************






 F1=HELP      F2=SPLIT     F3=END       F4=RETURN    F5=RFIND     F6=RCH
 F7=UP        F8=DOWN      F9=SWAP     F10=LEFT     F11=RIGHT    F12=RET
```

Figure 11. The DCE Configuration Screen

Underneath this entry menu there are more configuration screens.

On OS/390, the DCE administrative and user commands can be invoked from the TSO command line or, in most cases, they can be submitted as a job. The commands available from TSO/E or batch (in JCL) are:

DCECP The DCE Version 1.1 tool for administration of DCE. The dcecp command replaces most of the tools listed below, but they are still available for backward compatibility.

ACLEDIT Used to create, modify, and display Access Control List entries.

CDSCP The Cell Directory Service Control Program supports the CDS system administration task.

DTSCP The Distributed Time Service Control Program performs time configuration and management tasks for the distributed time service.

KDESTROY Destroys the login context of a principal and any associated credentials.

KINIT Obtains and caches the Kerberos Ticket-Granting Ticket and refreshes all currently held service tickets for DCE applications server.

KLIST Displays the principal names and passwords.

RGYEDIT The registry editor (rgy_edit) program provides facilities for creating and maintaining information kept in the registry database.

RPCCP The Remote Procedure Call Control Program provides a means of managing the name service database information and endpoints for RPC applications and their users.

The OS/390 UNIX shell offers the common OSF commands such as dcecp acl_edit, cdscp, dtscp, rgy_edit, rpccp, and so on.

3.1.5 Application Development

DCE on OS/390 is able to act as a DCE (RPC) client or server. Instead of creating a new application from scratch, it is also possible to port either the client or the server part of an existing DCE application to the OS/390 platform. This section explains the environment required to develop a basic client/server application. Section 3.1.6, “DCE Application Support for CICS and IMS” discusses a middleware product that enables access from a DCE client to existing (or new) CICS or IMS applications.

DCE is currently based on the usage of the C programming language. The Application Enablement optional feature for OS/390 is required for development of DCE client or server applications.

The application programmer can choose between the TSO or the OS/390 UNIX environment for the development. The OS/390 UNIX environment is identical to other UNIX-based DCE development environments. It is also possible to develop DCE applications from TSO, based on the facilities TSO provides.

3.1.6 DCE Application Support for CICS and IMS

OS/390 DCE Application Support (AS) provides CICS and IMS access to a class of users that have not been able to access those systems. It opens up CICS and IBM environments and makes them available to DCE clients on any platform. AS provides access to existing data and application logic by means of RPC. To the DCE client, AS appears to be just like any other DCE server. This maximizes the use of existing application-development skills at both the client and the server ends.

At the server side, existing, well-structured transactions can be accessed through RPCs with potentially few or no modifications to the existing CICS or IMS transactions. See 3.1.6.1, “Server Support” on page 35 for restrictions.

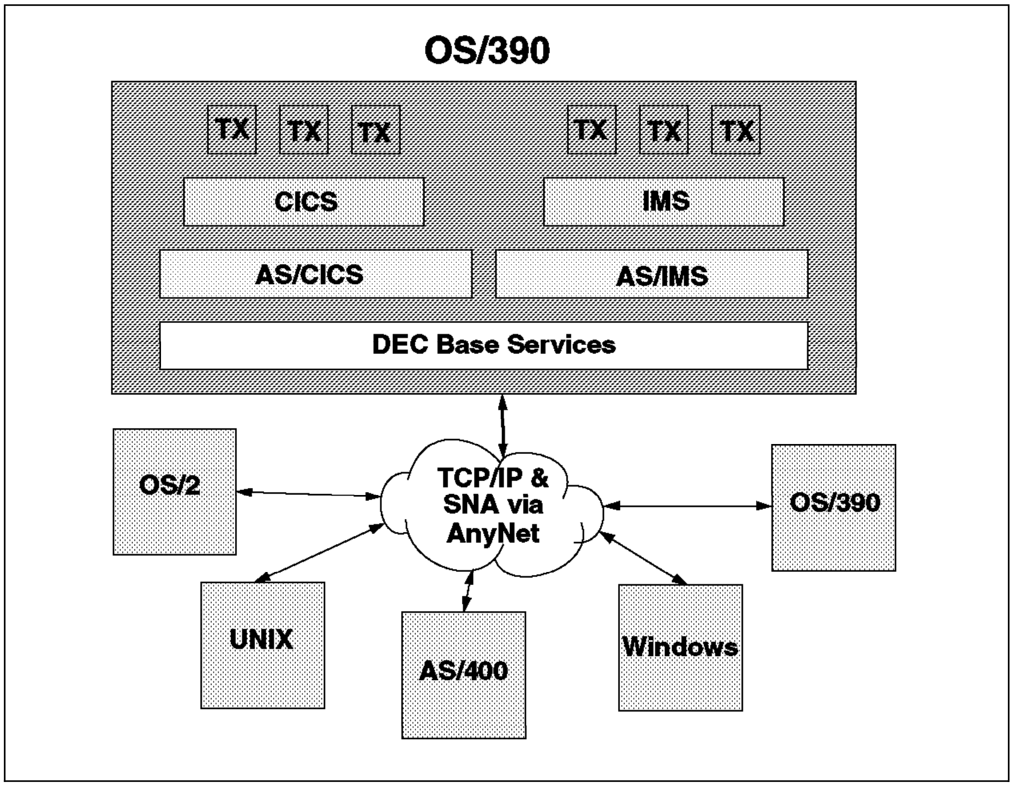


Figure 12. MVS/ESA OpenEdition Application Support Server Environment

AS/CICS allows access to CICS/ESA 4.1, and AS/IMS allows access to IMS/ESA Transaction Manager Version 3.1 or higher. Application Support provides enabling of Interface Definition Language (IDL) compiler extensions and a TSO ISPF-based administrative control program.

3.1.6.1 Server Support

A DCE client program can invoke CICS transactions that use the COMMAREA for input and output parameters or IMS transactions apart from message switches.

The CICS or IMS transactions programs can be written in any language and include, for example, transactions generated by CSP or VisualGen, provided that the data declarations can be described using COBOL data type extensions. Existing programs that satisfy these requirements, and do not contain terminal handling or BMS maps (basic mapping support in CICS), can be used without any change.

The host application programmer writes an extended interface definition language (XDL) file to describe which CICS or IMS transactions can be accessed and what the CICS or IMS programs are expecting to receive as input and give as output.

Once the XDL file is written, it can be compiled to produce the .cidl, .cacf, and .h files required on the client and server sides. The server stub is C compiled into a load module and placed into a stub load library as the server stub program. The client files are sent to the client and form the basis for client program development.

Application Support automatically handles the conversions of the COBOL and C data types.

3.1.6.2 Client Support

The client program uses RPCs to call a CICS or IMS application program through Application Support. The client does not need to know where the CICS or IMS programs are located. The underlying DCE Directory Service and DCE runtime provide location information.

Client programs are written in C language just like any other DCE client program. Although the DCE portion of the client program should be written in C, there is nothing that prevents you from using other languages or visual tools, for example, to build the Graphical User Interface (GUI) and main application code.

If you are writing client programs, you do not have to understand either CICS or IMS because the DCE portion of the client program uses C and standard DCE APIs.

When converting an existing application, the client programmer obtains an IDL file created by the host application programmer, as explained in 3.1.6.1, “Server Support” on page 35. The programmer will compile the IDL file on his/her platform and will also design and implement the client program logic and user interface on the basis of the host transaction descriptions. The calls they can use are described in the interface definition file.

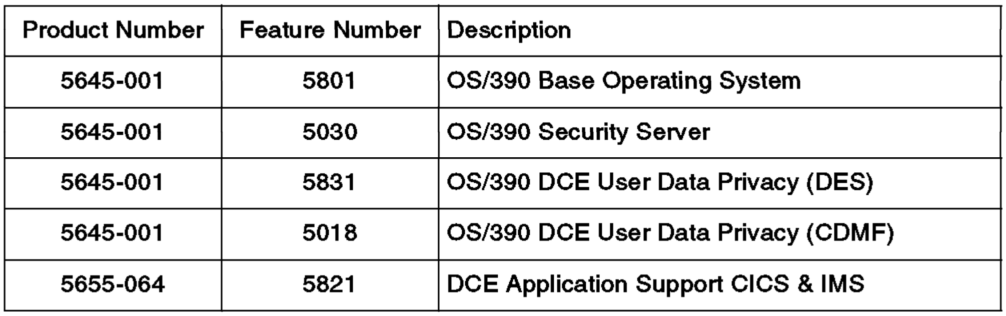
3.1.6.3 Security Implementation

The main purpose of Application Support is to make CICS and IMS transactions accessible to DCE clients. CICS and IMS transactions live in an environment protected by RACF. RACF authenticates users and authorizes resource access based on access lists defined for each resource. The RACF and DCE security integration provides the mechanism for mapping the RPC authentication information to resources protected by RACF.

To summarize, the combination of DCE and Application Support performs authentication, and RACF performs authorization.

3.1.7 Products Packaging and Prerequisites

The following table gives an overview on order numbers of the different DCE products available for OS/390.



Product Number	Feature Number	Description
5645-001	5801	OS/390 Base Operating System
5645-001	5030	OS/390 Security Server
5645-001	5831	OS/390 DCE User Data Privacy (DES)
5645-001	5018	OS/390 DCE User Data Privacy (CDMF)
5655-064	5821	DCE Application Support CICS & IMS

Table 2. IBM DCE Products for OS/390

3.2 VM/ESA OpenEdition Implementation of DCE

VM/ESA OpenEdition Environment

VM/ESA OpenEdition refers to services offered with VM/ESA Version 2 that provide industry standard interfaces for applications and users. It consists of three main parts:

- OpenEdition VM Services with the POSIX 1003.1 compatibility functions (as part of the base VM product)
- OpenEdition VM Shell and Utilities (A feature with a small charge. Though not required for DCE, it allows a more complete access to the Byte File System where DCE data is stored, and better control of the multitasking POSIX environment.)
- OpenEdition VM DCE (A no-cost base feature and a restricted data packet privacy feature)

3.2.1.1 Processes and Threads in VM/ESA OpenEdition

Virtual machines are tied together by the Control Program (CP) and are very well insulated against each other so that complete guest operating systems, such as CMS, MVS, or VSE, can be run in complete isolation on the same CP.

A CMSCALL from anywhere to a multitasking program results in the creation of a new CMS process, and all POSIX programs are treated as multitasking programs. A CMS process becomes a POSIX process whenever the first POSIX kernel function is invoked in it. A POSIX process is really just a CMS process with enough extra structure to support the POSIX kernel services, such as signals.

The `spawn()` POSIX system call (a proposed POSIX standard) provides a fast, low-overhead way of creating a new process in the same CMS address space. The `fork()` system call, which clones an existing address space to form a new one, is not supported.

VM/ESA OpenEdition provides support for the OSF DCE threads API. These DCE threads are implemented as POSIX threads, which in turn are mapped to CMS threads in the CMS multitasking base. This provides the multithreading support required by the DCE kernel, applications, and utilities.

POSIX threading, as implemented on VM, is based on the POSIX.1c standard. However, OSF′s implementation of DCE uses Draft 4 of the standard. CMS′s implementation of POSIX threads is at the Draft 6 level. The DCE threads implementation on VM solves the incompatibilities through the use of a header file and library routines that adjust the Draft 4 level expected by DCE to the Draft 6 level expected by POSIX.

Since an application program can use the DCE APIs, the POSIX APIs, or native CMS APIs, the programmer must decide which thread API to use; for example, they can use `pthread_create()` (DCE) or `ThreadCreate()` (CMS). These calls even have differences between the drafts. One draft expects a value as a parameter, the other draft expects a pointer to a value as the parameter. As a result of compatibility and portability issues, it is required that DCE applications use only DCE threads.

3.2.1.2 Byte File System

To support POSIX.1 and .1a standards on VM/ESA OpenEdition, a hierarchical file system called the Byte File System (BFS), which is also used by DCE, was created. BFS data can be stored in the same server as Shared File System (SFS) data. The SFS is VM's native hierarchical file system. The SFS server keeps its data in a file pool, which is made up of one or more storage groups. At the file pool and storage group levels, BFS and SFS data can coexist.

Although BFS data resides within a SFS server, it uses its own rules and authorizations. VM/ESA OpenEdition maintains in the CP directory POSIX user and group identifications through the assignment of user and group IDs (UIDs and GIDs), as in UNIX, for each CMS user. Therefore, authorizations to files are based on a UNIX model, not the conventional SFS CMS authorizations.

BFS data is represented differently from the record-oriented CMS approach. A POSIX-compliant application sees BFS data as a stream of bytes and defines the record structure itself by inserting special control characters into the data stream. The POSIX file system structure is the same as a traditional tree-structured hierarchical UNIX file system, with a root directory containing files and further directories. From a VM perspective, SFS first introduced the concept of paths and directories to CMS, but still retained the concept of the file type and file mode. POSIX does not use this concept. The introduction of BFS extends this concept to make BFS function like a UNIX file system.

The BFS has the same flexibility as the UNIX file system. BFS file spaces are created, then mounted explicitly or implicitly at specific mount points. For example, in order to use the BFS, at least the *root* file space must be mounted. Typically, this is for the read-only programs, control files, and samples. Then the read/write user file space can be mounted at a preselected mount point.

Figure 13 shows an example that explains this idea.

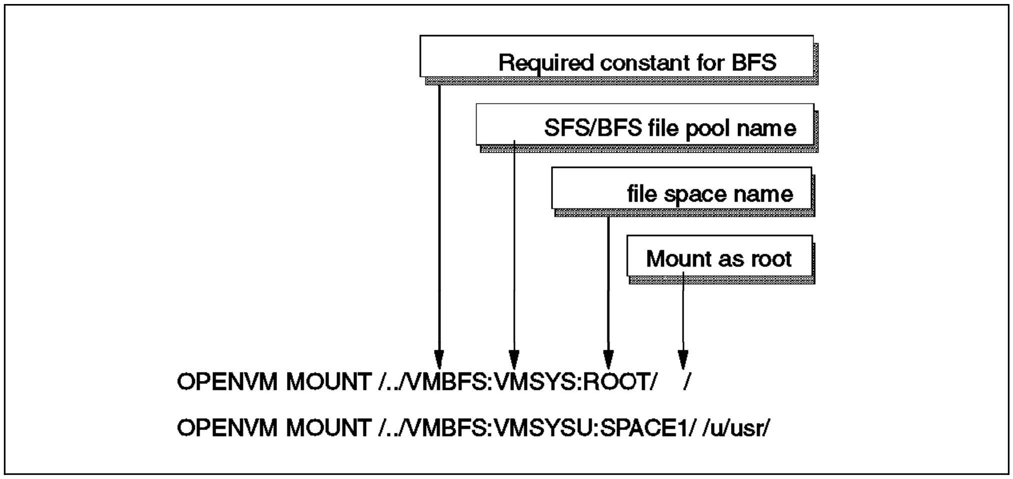

Required constant for BFS
SFS/BFS file pool name
file space name
Mount as root
OPENVM MOUNT /../VMBFS:VMSYS:ROOT/ /
OPENVM MOUNT /../VMBFS:VMSYSU:SPACE1/ /u/usr/

Figure 13. Mount Points

Issuing the commands shown in the figure first mounts the /../VMBFS:VMSYS:ROOT/ file space (*root*) and all its directories, which includes the */u* (a typical alias for */home*) directory. This BFS is in the generic VMSYS system file pool that is created as part of a standard VM system installation. The second mount is for a file space that was enrolled as *SPACE1* in the VMSYSU file pool. This effectively connects the */u* with the */usr* that would have been created in SPACE1. Using this approach, DCE for VM/ESA creates the file-tree structure shown in Figure 14 on page 39.

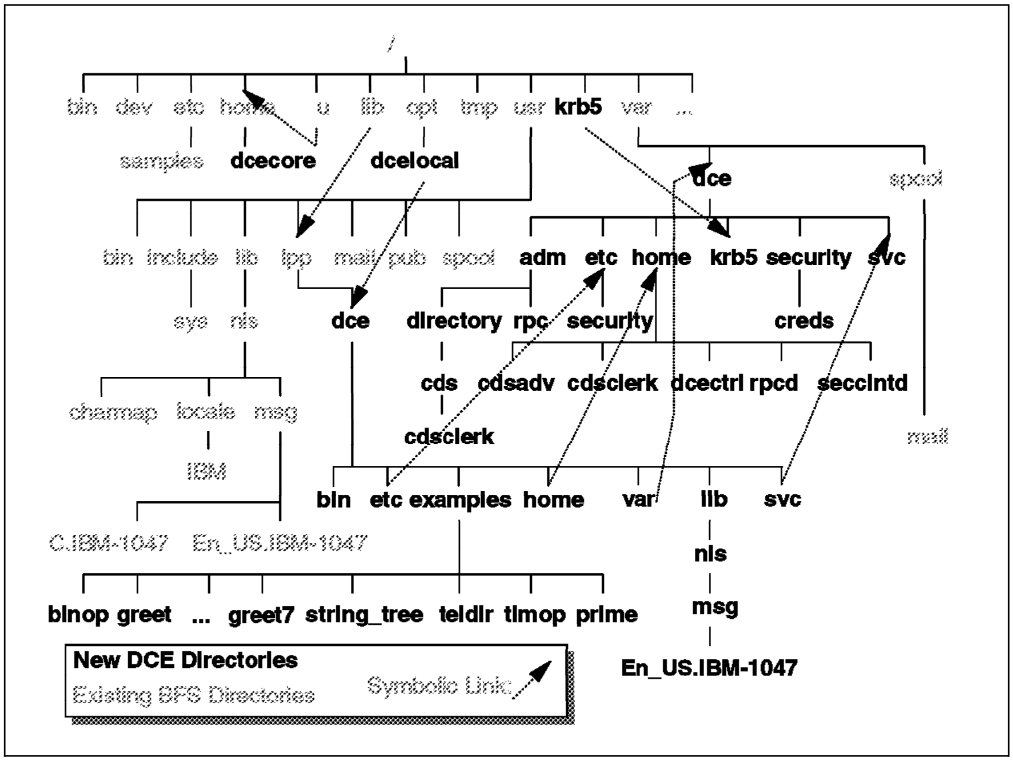

/
bin dev etc home u lib opt tmp usr krb5 var ...
samples dcecore dcelocal
dce
spool
bin include lib lpp mail pub spool
adm etc home krb5 security svc
sys nls
dce directory rpc security
creds
cds cdsadv cdsclerk dcectrl rpcd secclntd
charmap locale msg
cdsclerk
mail
IBM
bin etc examples home var lib svc
C.IBM-1047 En_US.IBM-1047
nls
binop greet ... greet7 string_tree teldir timop prime
msg
New DCE Directories
Existing BFS Directories
Symbolic Link
En_US.IBM-1047

Figure 14. Byte File System Tree Structure

VM/ESA OpenEdition allows DCE programs to simultaneously access POSIX BFS and CMS record-oriented data within the SFS repository. In addition, it allows an external link to be defined that allows a logical BFS file to represent a file in a CMS file space. DCE uses external links so commands and configuration data can be located in one place and shared by both CMS and POSIX shell users.

Unlike SFS users, which generally have direct ownership for one, and only one, SFS file space, BFS clients need not have this one-to-one relationship to a particular BFS space. In another words, there is no implicit file pool associated with BFS file space. BFS extends the flexibility establishing the relationship between the BFS clients and BFS file spaces. A file space is an administrative control point for managing space consumption by client′s that write into the file space. The file space is created with the command ENROLL USER, establishing a number of logical blocks that limit file creation and extension in that file space.

3.2.1.3 OpenEdition Shell and Utilities

The OpenEdition Shell and Utilities, a POSIX 1003.2 implementation, is a separately priced feature of VM/ESA Version 2. It is based on functions from the Bourne shell and UNIX Korn shell.

The shell is a comprehensive programming environment with tools that allow you to create and port DCE programs from UNIX environments more effectively. Make files, shell scripts, and powerful tools such as grep and awk are just a part of what the shell offers.

By invoking the openvm shell command, a CMS user initializes the shell environment. The user may enter shell commands, address CMS and CP to invoke commands from that environment, use XEDIT on Byte File System files, or even invoke a second nested shell.

A point to keep in mind is that the OPENVM commands only contain a subset of the functions that are provided through native shell commands. For example, you cannot control if a job is running in the foreground or background from CMS. It is also much easier to terminate jobs already running with signals from inside the shell.

To give a CMS user a sample of POSIX commands, shown in Table 3 is a list of some CMS, OPENVM, and shell commands. The CMS commands are listed only to give someone new to a UNIX shell some point of reference. It is the OPENVM commands that actually perform similar operations.

CMS Command	OPENVM Command	Shell Command	Description
GRAnt AUThority	OPENVM PERMit	chmod	Change access permissions to a file
CREate DIRectory	OPENVM CREate DIRectory	mkdir	Create directory
ACCess	OPENVM MOUnt	cd	Change the working directory
COPYfile	OPENVM PUTbfs	cp	Copy a file
ERASE	OPENVM ERASE	rm	Erase a file or directory
Listfile	OPENVM LIStfile	ls	List files within current directory
LOAD	LOAD	ld	Load a file
LOGoff	LOGoff	exit	Exit the system, or shell
Q ACCESSED	OPENVM QUERY MOUNT	pwd	Return the working directory
TXTlib/TAPE	OPENVM PARchive	ar	Create, dump, and maintain libraries

Table 3. CMS and POSIX Commands Comparison

3.2.2 DCE Implementation Specifics

OpenEdition DCE is implemented on VM in three major forms, namely:

- Daemon Processes

  These processes provide DCE services for all DCE users on a particular VM system. These processes run as individual POSIX processes within a dedicated virtual machine called DCECORE. A control task process running on DCECORE initiates and controls the execution of the daemon processes and provides their automatic restart if daemon processes fail. An administration utility, DCECTRL, allows the monitoring and controlling of the daemon processes from virtual machines other than DCECORE.

  The daemons running in the DCECORE virtual machine are:

  RPC daemon Supports the use of RPC services on a host and provides the host′s endpoint map service

Security client Communicates with security servers in a cell to request information and perform operations using the registry database

CDS advertiser Solicits and advertises the names and the statuses of all CDS servers' databases in the network

CDS clerk Processes all naming requests from DCE clients and is the interface between client applications and CDS servers

DCE Control Task Initiates, terminates, and oversees the other daemons

Figure 15 shows the interactions and implementation of the DCE components on VM/ESA.

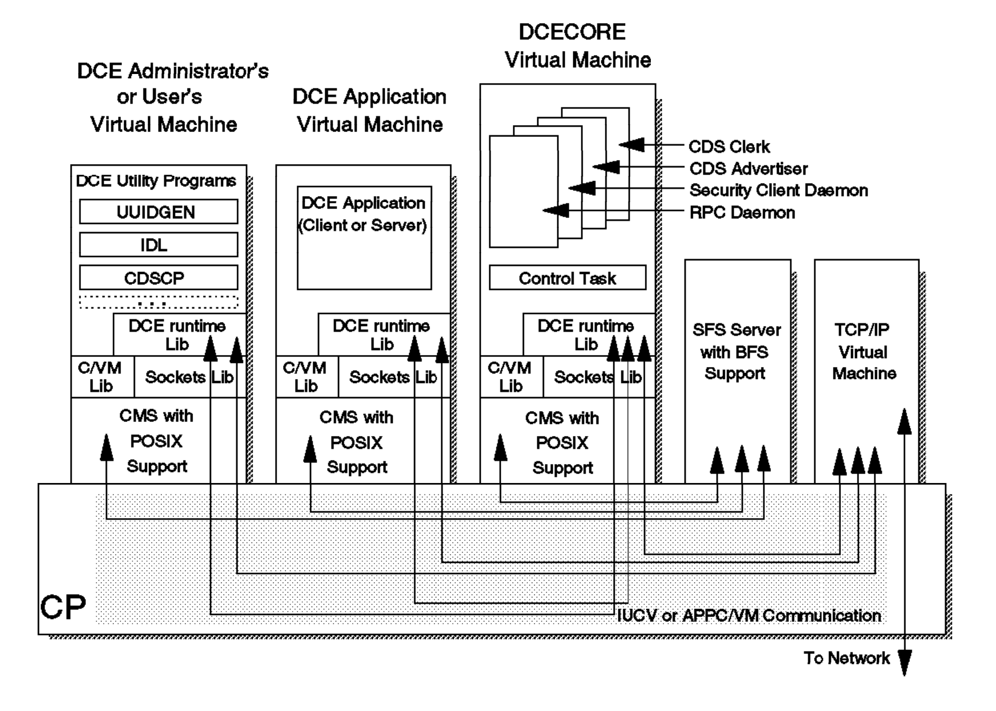


Figure 15. VM DCE Implementation

The DCE daemons are located in the DCECORE machine managed by a control task. The administration utilities, or application programs communicate with the DCECORE machine using the Inter-User Communication Vehicle (IUCV). The TCP/IP machine provides an Internet Protocol (IP) connection to the over host in the DCE cell. The SFS server configured as a BFS server is used by applications, utilities, and DCECORE as a program and data repository.

- Runtime Support

This is a library of APIs that support the DCE applications, services, and utilities. This library usually resides in the EUVSEG shared segment linked by the virtual machine running a DCE application, daemon process, or utility program. This code provides a collection of C language functions used by DCE applications, daemons, and utility programs.

- Utility Programs

  These are used for developing DCE applications and administering DCE. These utility programs run as POSIX programs within the application developers′, administrators′, or users′ virtual machines. Figure 16 shows the various DCE utilities in an abstract pictorial format.

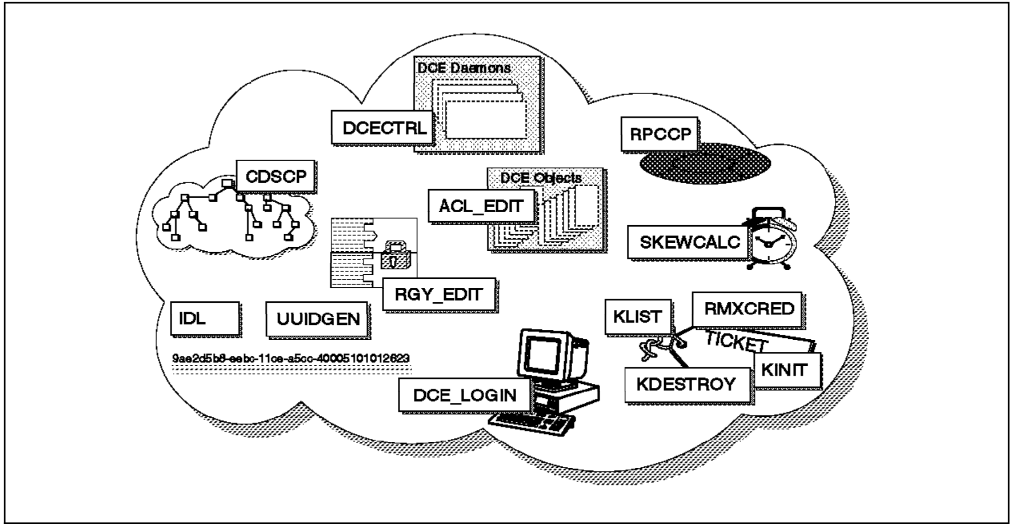


Figure 16. VM DCE Utilities

3.2.2.1 Remote Procedure Call

OpenEdition DCE fully supports DCE RPC, making VM DCE applications and data available to your distributed computing environment. The RPC daemon (RPCD) is the server process that supports the use of RPC services on the host.

For a RPC client program to send a RPC request, the client must be able to locate the host with a RPC server that can receive the request and the address (endpoint) of the process on that RPC server. The client uses the directory service to find an appropriate server and the host local RPCD to locate the required process or interface on that server. Every DCE host machine must run a RPCD. RPCD provides the endpoint map service, which enables a client to locate the network endpoint or port of the process that it requires. RPCD′s endpoint map service enables RPC clients and RPC servers to communicate without the need to know where the interface endpoint is. The endpoint map is simply a file that exists in the BFS. The RPCCP utility is used by an administrator to maintain the endpoint map, determine the availability of remote servers, and perform tasks related to naming and location of servers.

3.2.2.2 Directory Service

VM/ESA′s implementation of DCE directory services is based on the X.500/ISO 9594 international standard for intracell (within a cell) directory services. The DCE directory service provides services at the cell and global levels.

Cell Directory Service: OpenEdition DCE implements the CDS advertiser and CDS clerk. The CDS server must be running on some other machine in the cell. OpenEdition DCE also supports the CDS Control Program (CDSCP), a utility to control CDS services on any host within a DCE cell.

The CDS Advertiser is responsible for sending and receiving information or advertisements of the presence and status of CDS servers on DCE hosts. It is also responsible for creating the cache used by the CDS clerk and periodically saving the cache to the disk.

The CDS clerk is running on a host that is not running a CDS server. It serves as the intermediary between client applications and CDS servers. Its main function is to maintain a cache of information obtained through directory queries. The clerk stores the responses to the queries in its cache; so the next time a similar query is requested, the information is already available to the client, avoiding the need to communicate across the network to the CDS server.

Global Directory Service: OpenEdition DCE does not implement GDS. If inter-cell name resolution is required, GDS and GDA must be installed and administered from a non-VM host in your cell. OpenEdition DCE for VM/ESA also uses the name-to-address translation services of the Domain Name System (DNS). DNS is used by the Internet to keep track of the number of machines that comprise the Internet. DNS provides an alternative for independent cells to locate and interact with each other.

3.2.2.3 Security Service

DCE on VM provides a security client daemon that provides access to all security services provided by a non-VM security server. A security server is not currently provided on VM. The Registry Service editor (`rgy_edit`) and the Access Control List editor (`acl_edit`) are available with OpenEdition DCE for VM/ESA, so you can administer an entire DCE cell security registry from VM, through the use of the security client daemon.

3.2.2.4 Distributed Time Services

In many DCE implementations, the DCE DTS daemon is responsible for keeping the clock of the local host within the five minute tolerance required for DCE login and authenticated RPCs to function. However, VM/ESA Version 2 does not implement the DTS daemon. Instead, it provides a time skew file as an alternative method for allowing relatively coarse compensation to the clock differences between VM and the DCE cell time. A routine called `skewcalc`, computes the time difference between the VM/ESA host and a reference system supporting the Internet Time protocol, and places the delta in a time skew file in the BFS that is used by the authentication and login services. This adjustment does not affect the time-of-day values that application programs may require for sequence critical transaction processing. But, it does allow system administrators a way to let VM coexist in a highly available environment where the cell time will slowly drift away from CP′s time.

Figure 17 on page 44 shows the interactions of the DCE time service and the `skewcalc` utility.

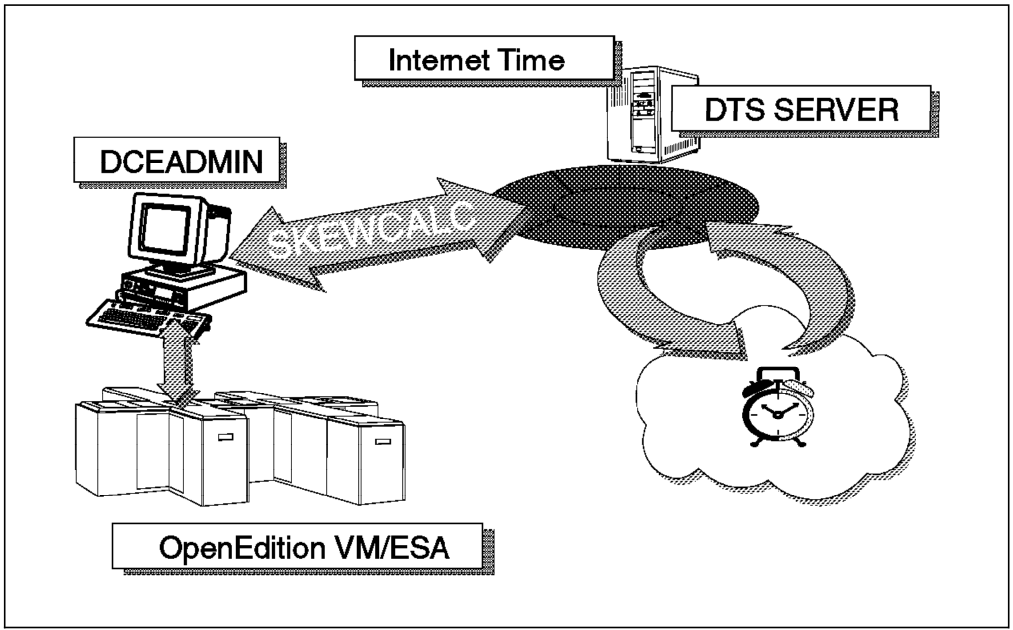


Figure 17. VM DCE Time Implementation

Fully supported on VM are the DTS APIs. The DTS APIs provide routines that applications can use to obtain and manipulate binary time stamps that are based on Coordinated Universal Time. The DTS API supports ANSI C language constructs.

3.2.3 User Environment

In the current releases, VM security (RACF) and DCE security are not integrated. This section discusses how users can get access from one environment to the other.

3.2.3.1 VM/CMS User Access to DCE

After a CMS user has logged in to CMS, he/she can work in CMS or call the POSIX shell (`OPENVM SHELL` command) if this optional feature is installed. In order to access the POSIX shell, the user must have a UID/GID assigned. The UIDs and GIDs are implemented in the CP directory, which also defines the virtual machines (users) and characteristics. The POSIX shell enables the user to work in a UNIX-like environment and with the Byte File System, but they are still local users only, with no authenticated DCE cell access.

To get access to DCE, the CMS users must to be defined in the DCE security registry. Once they are logged on to a VM environment, they log on to DCE in one of the following ways:

- `DCELOGIN` (from CMS or the shell)
- `dce_login` (from CMS or the shell)

After this step, they can run DCE application clients as authenticated principals. The login context created by the dce_login is maintained until the user logs off VM, IPLs CMS, or runs the `kdestroy` DCE utility.

3.2.3.2 VM Access from DCE Clients

When DCE clients access a server on VM, they are not completely isolated from the native security provided by VM. These DCE server processes are started beforehand and run with the ESM (External Security Manager, such as RACF) and CP privileges of the VM users who started the server processes. It is up to each DCE server process to check the permissions of the incoming client principal, based on their name, UUID, group memberships. If an application implements an ACL manager, the permissions can be administered externally.

An incoming DCE client does not have free access to VM. The access is restricted to the services a DCE server offers and beyond that to the ESM/CP privileges of the VM user who started the server daemon (listening process). If a virtual machine provides many services, you must be aware that any task running inside a virtual machine shares a common address space with other tasks. It is without doubt that a clever individual left with unrestricted access could obtain all data and privileges available to the entire virtual machine and those of the tasks running. The key message is never give more than the minimum CP, ESM, and DCE privileges required for a specific server.

3.2.4 Administration Tools

DCE administration on VM/ESA is provided through a set of stand-alone utilities, as follows:

IDL Interface Definition Language, an RPC interface in a high-level C-like declarative language

UUIDGEN UUID generator

RPCCP RPC control program that performs RPC administration tasks

CDSCP CDS control program that performs CDS administration tasks

ACL_EDIT Permission utility that edits the Access Control Lists of DCE objects

RGY_EDIT Registration utility that maintains the Security Registry

DCE_LOGIN Process that initializes a user′s DCE Security environment, and authenticates the user to the security service

KINIT Command that obtains and caches the ticket-granting ticket

KDESTROY Command that destroys a principal′s login context and associated credentials

KLIST Command to list cached tickets

VM includes additional utilities that assist in managing the VM DCE environment. They are:

DCECTRL A command that controls the status of the DCE daemons

RMXCRED A utility that removes stale credentials from the BFS

SKEWCALC A sample utility to import DCE cell time into VM for use with VM DCE services.

These utilities are accessible from inside the OE (OpenEdition) Shell, or directly from CMS. Utilities containing a underscore in their names, such as acl_edit, can be used without the underscore if entered from the CMS command line. All commands must be entered in lowercase while inside the POSIX shell.

3.2.5 Application Development

DCE on VM is able to act as a DCE (RPC) client or server. Application servers run in a CMS virtual machine, as depicted in Figure 15 on page 41.

Instead of creating a new application from scratch, it is also possible to port either the client or the server part of an existing DCE application to the VM platform. This section explains the environment required to develop a basic client/server application.

DCE is currently based on the usage of the C programming language. Therefore, an appropriate C compiler is needed. The IBM C for VM/ESA compiler implements the ANSI C standard and ISO C standard and is compatible with the DCE requirements. The C-Runtime and Language Environment is extended to include the POSIX functions.

The `uuidgen` utility is used to create a new IDL file prototype with a new UUID. If you port an application to VM, you take the IDL file that already exists instead of running `uuidgen`. Once the IDL file is completed, the IDL compiler is used to precompile it to create the header and stub files used by the client and server program.

Application development can be distilled into the following steps in VM:

1. Generate a UUID and IDL template.
2. Name the interface for your application.
3. Define the interface operations.
4. Compile the interface with the IDL compiler. Input and output files can be CMS or BFS.
5. Write server and manager code for your application.
6. Write the client code.
7. Compile the client and server programs via the `c89` utility, and be aware that:
   - The POSIX headers must be in the search path.
   - The sockets and DCE headers are located on the S-disk (MAINT 190).
   - Input and output files can be CMS or BFS.
8. Build your application in one of the two following ways:
   - Link-edit your application with DCE TCP/IP and LE runtime libraries via the `c89` utility.
   - Use `make` under the shell.
9. Applications must have POSIX(ON).

   POSIX applications can run in a shell environment as well as in CMS. When running a POSIX application as a CMS module, you must invoke it using the `OPENVM RUN` command. When running in CMS, you need to specify the Language Environment run-time option POSIX(ON) to request that your application run as a POSIX application. DCE applications must run as POSIX(ON) applications because the DCE run-time library functions require the use of POSIX services.

As of today, there are no tools available that offer a higher-level API to DCE or that would generate part of a DCE application server or client. Tools, such as Encompass or other tools mentioned in Chapter 7, “Client/Server Application Development” on page 107, are not available on the VM platform.

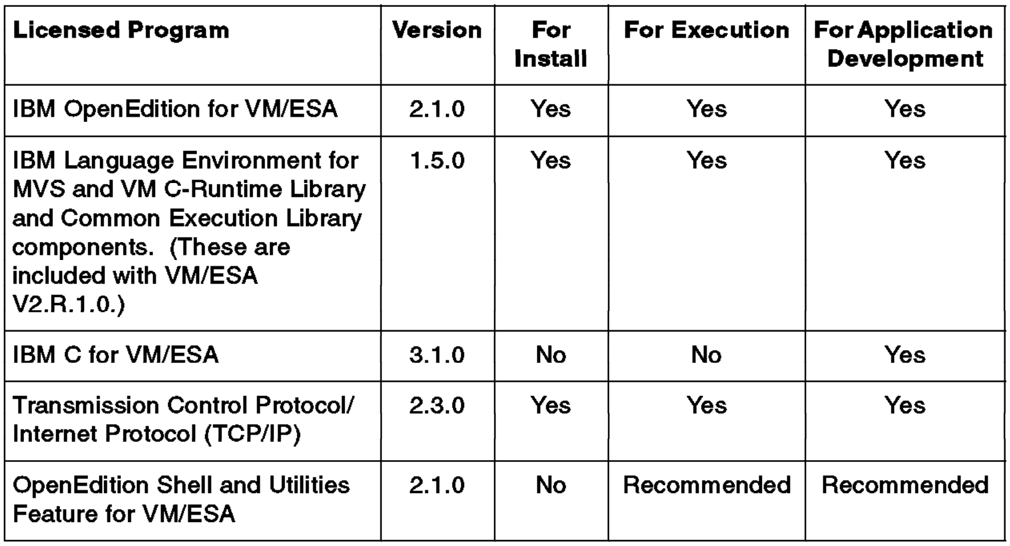
Client applications can display their output in many ways. A partial list includes:

- Be character oriented for use with a 3270 terminal connected to CMS.
- Be an X-Windows application running under CMS with a window running on any X-Windows server workstation.
- Use the new CMS GUI APIs for AIX, Windows, or OS/2 Warp remote presentation.

The CMS GUI is part of VM/ESA Version 2 and comes with CMS Desktop supporting regular CMS functions and a GUI API for support of new applications.

3.2.6 Products Packaging and Prerequisites

DCE features, as well as the POSIX support features (threads, APIs, BFS), will be shipped with VM/ESA Version 2 at no additional cost. To enable OpenEdition DCE for VM/ESA, you need the following:



Licensed Program	Version	For Install	For Execution	For Application Development
IBM OpenEdition for VM/ESA	2.1.0	Yes	Yes	Yes
IBM Language Environment for MVS and VM C-Runtime Library and Common Execution Library components. (These are included with VM/ESA V2.R.1.0.)	1.5.0	Yes	Yes	Yes
IBM C for VM/ESA	3.1.0	No	No	Yes
Transmission Control Protocol/ Internet Protocol (TCP/IP)	2.3.0	Yes	Yes	Yes
OpenEdition Shell and Utilities Feature for VM/ESA	2.1.0	No	Recommended	Recommended

Table 4. Prerequisites for OpenEdition DCE for VM/ESA

VM/ESA can only be integrated into an existing DCE cell. Because VM/ESA does not support the security server, CDS server, and DTS server, you should have them running and tested prior to the VM/ESA DCE installation. The VM/ESA clerks and clients must communicate with their respective servers in order to make the definitions required to include a VM host within a cell.

VM/ESA Version 2 requires an IBM ESA/370 or ESA/390 processor. It can also run on a PC Server 500 System/390.

3.3 OS/400 Implementation of DCE

DCE Base Services/400 enables the OS/400 to participate in DCE as a client machine. The current DCE implementation on OS/400 is based on OSF DCE 1.0.2. It provides full client and server support for RPC as well as client support for DTS, CDS, and security services. Interoperability with other DCE environments based on newer OSF implementations, such as OSF DCE 1.1, has been tested. For the time being, the CDS and security servers, which are mandatory in a cell, need to be running on another platform, such as IBM AIX.

The current releases of OS/400 are V3R2 (on AS/400 native CISC hardware) and V3R7 (on PowerPC-based RISC hardware). Although they have different version numbers, they differ only in their hardware support. All references to OS/400, DCE/400, and other software products and features in this section equally refer to both V3R2 and V3R7, respectively.

Although not formally announced, IBM representatives have stated in public that the following features will be supported in future releases:

- DTS, CDS, and security servers (formal Statement of Direction)
- DFS client
- OSF DCE Level 1.1

3.3.1 OS/400 Environment

In this section, we describe the features of the Common Programming API Toolkit/400 Feature (CPA) (feature #1690) that allow DCE to run on OS/400.

Figure 18 depicts the CPA as a collection of programming interfaces or functions that are modelled after a subset of functions in the Standard UNIX Specification or POSIX standards. A combination of services from OS/400 and CPA together provide the functionality required by DCE. For explanations on POSIX standards, see 3.1.1, "OS/390 UNIX Environment" on page 22.

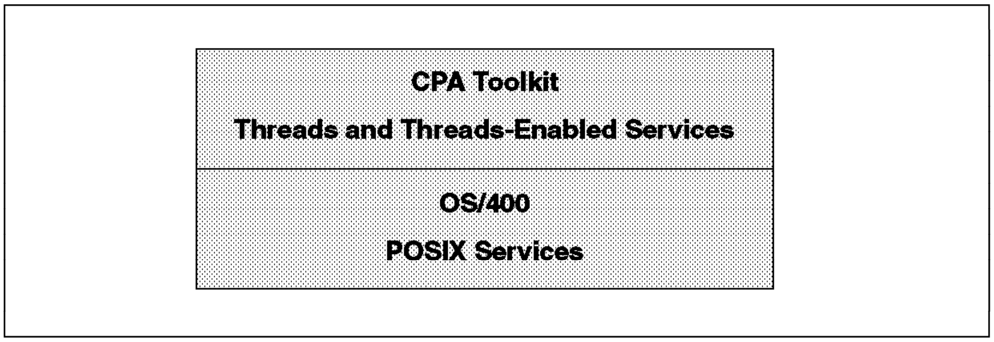


Figure 18. Common Programming Application Interface

The CPA provides support for:

- Threads based on POSIX draft standard 1003.4, draft 4a and the Open Software Foundation DCE Version 1.1 Application Development Reference.
- Thread synchronization using mutexes and condition variables along with thread-specific data.
- A thread-enabled C-runtime.
- Thread-enabled sockets based on Berkeley Software Distribution (BSD) sockets.
- Environment variables based on Spec 1170.

OS/400 provides support for:

- Thread-enabled stream file I/O.
- Semaphores based on Spec 1170 and the AT&T UNIX System V Interface Definition.

- Shared memory based on Spec 1170 and the AT&T UNIX System V Interface Definition.

3.3.1.1 Processes and Threads in OS/400 CPA

In the CPA implementation, multiple OS/400 jobs share the same program storage or ILE (Integrated Language Environment) activation group. In the CPA model, each OS/400 job that is sharing the ILE activation group is called a thread. The collection of all OS/400 jobs sharing an activation group is called a process. All of the threads in the process share the same global static storage and heap storage. Each thread has its own program stack and automatic storage.

From a UNIX point of view, there is no change for the terminology; all the properties are available, such as user space, system threads, system calls, and so on.

From the OS/400 point of view, there is no change either; only the names for the existing objects on OS/400, such as jobs (threads), process (set of threads that share an activation group), and so on, have changed.

3.3.1.2 Hierarchical File System

The Hierarchical File System on OS/400 is provided through the Integrated File System (IFS), which provides a collection of C language APIs to access stream files on a UNIX file system. The Hierarchical File System can be used by DCE applications.

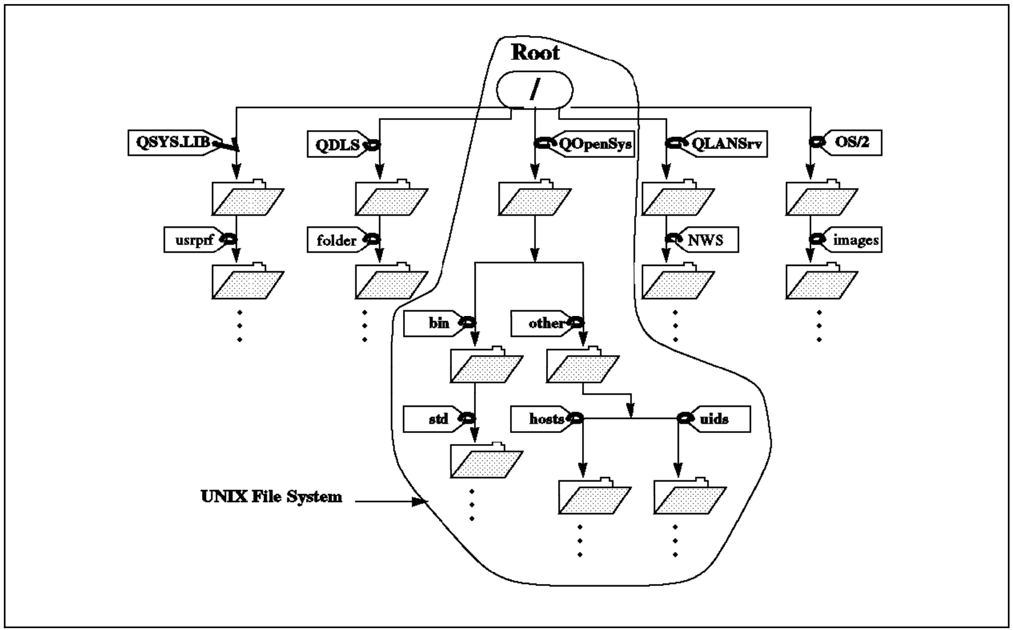


Figure 19. Integrated File System

The Integrated File System is a part of OS/400 that supports stream input/output and storage management similar to personal computer and UNIX systems, while providing an integrated structure over all information stored in the AS/400.

Generally speaking, a file system provides structures and methods to organize, store and access data in files, directories, libraries, and objects (in AS/400 terminology) on a computer′s permanent storage. These structures and methods may be different from one type of operating system to another. In fact, on OS/400, there are several types of file structures and access rules that can be thought of as file systems:

- Libraries storing database files and other objects
- Folders storing documents
- LAN Server/400 storing data for OS/2 LAN server clients

IFS does indeed treat these (traditional) types of file management on OS/400 as separate file systems. IFS defines the file tree, as shown in Figure 19 on page 49, with the following top-level directories:

root The root (/) file system is designed to take full advantage of the stream file support and hierarchical directory structure of the Integrated File System. It has the characteristics of the DOS and OS/2 file systems.

QOpenSys The open systems file system is designed to be compatible with UNIX-based open system standards, such as POSIX and XPG. Like the / (root) file system, it takes advantage of the stream file and directory support provided by IFS. In addition, it supports case-sensitive object names.

QSYS.LIB The library file system supports the OS/400 library structure. It provides access to database files and all of the other OS/400 object types that are managed by the library support.

QDLS The document library services file system supports the folders structure. It provides access to documents and folders.

QLanServ The LAN Server file system provides access to the same directories and files that are accessed through the LAN Server/400 licensed program. It allows users of the OS/400 file server and OS/400 applications to use the same data as LAN Server/400 clients.

Users and application programs can interact with any of the file systems through the IFS interface by specifying the correct path name. For instance /QSYS.LIB/usrprf/... would access a file in the usrprf library in QSYS.LIB. However, this interface is optimized for input/output of stream data, in contrast to the record input/output provided through the data management interface. While IFS allows access to *all* files, the traditional data management interface does not provide access to QOpenSys or stream data files.

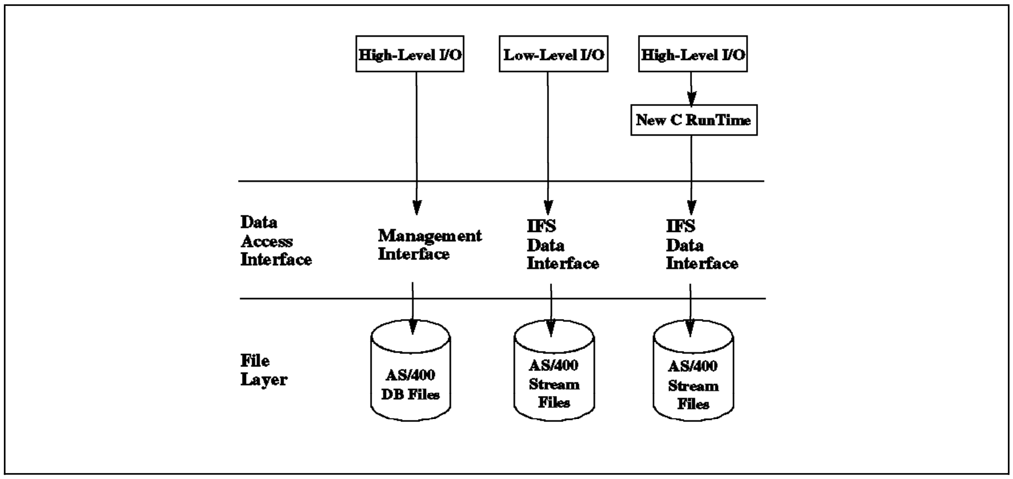


Figure 20. AS/400 Access Method

High-Level I/O in Figure 20 means using buffered stream I/O routines, such as `fopen()`, `fread()`, `fwrite()`, `fseek()`. Low-Level I/O means using `open()`, `read()`, `write()` POSIX calls, which are system calls in UNIX. When using the High-Level I/O interface from ILE C/400, the program accesses AS/400 DB file through the management interface, whereas Low-Level I/O automatically uses the IFS interface. To use the High-Level interface for IFS access, the programmer needs to explicitly specify that at compile time.

The QOpenSys file system is the UNIX file system on OS/400. The UNIX file system is fundamentally different from the traditional AS/400 file system that was (and is) an integrated relational database file system. The QOpenSys file system was designed to provide the benefits of a server environment, portability and interoperability, to a UNIX client.

Some features of QOpenSys file system are:

- Accessed through a hierarchical directory structure similar to UNIX systems
- Optimized for stream file input/output
- Supports multiple hard links and symbolic links
- Supports case-sensitive names
- Supports local sockets

3.3.1.3 Shells and Utilities

POSIX 1003.2 shells and utilities are not yet available on OS/400. However, there is a Statement of Direction to implement this in the future.

3.3.2 DCE Implementation Specifics

AS/400 supports the DCE architecture via DCE Base Services/400, which is based on OSF DCE Version 1.0.2. The following components, shown with their associated processes (daemons), are implemented:

- RPC client and server (RPC daemon)
- Security service client (security client daemon)
- CDS client (CDS advertiser daemon and CDS clerk daemon)
- DTS client (DTS daemon)

DCE Base Services/400 offers all APIs to access CDS, security and DTS servers, but does not implement these services on the AS/400 platform. Support of RPC client and server allows for development of full DCE application clients and servers on OS/400. However, only connectionless RPC protocol is supported.

The Distributed Time Services (DTS) clerk daemon synchronizes, adjusts, and maintains the DCE time of the local machine with the values obtained from DTS servers in a distributed network. However, on the AS/400, the DTS daemon does not update the hardware system clock, because it cannot be changed. Therefore, CPA maintains a separate software clock that can be adjusted and used by DCE.

The OSF DCE code is enhanced by DCE Base Services/400 to provide the familiar look and feel of AS/400 with support for AS/400 messages, menus, prompts, and help text. In addition, enhancements are also added to increase reliability, availability, and serviceability of the DCE code to make it consistent with other AS/400 products.

The following table depicts how the various functions are implemented in the components as a base for DCE:

Function	OS/400	ILE C/400	CPA Toolkit
Hierarchical File System	✓		
BSD Sockets	✓		
Security Extensions	✓		
ANSI-C		✓	
Semaphores			✓
Threads			✓
Shared Memory			✓
C-Language API to IFS			✓

Table 5. DCE/400 Base Function Implementations

3.3.3 User Environment

The user environment has not changed because of DCE. OS/400 security and DCE security have not been integrated. This section discusses how users can get access from one environment to the other.

3.3.3.1 OS/400 User Access to DCE

If users want to get their DCE network credentials, they have to log in as normal OS/400 users and then use the `DCE_LOGIN` or the `STRDCESSN` command, for which they need to provide a DCE principal name and a DCE password.

Every DCE user has an account in the registry database. The account identifies the user′s name and password. This password is used in the authentication of user accounts.

The user can also use the commands or menus available via DCE/400, such as `klist`, `kdestroy`, `kinit`, and so on.

3.3.3.2 OS/400 Access from DCE Clients

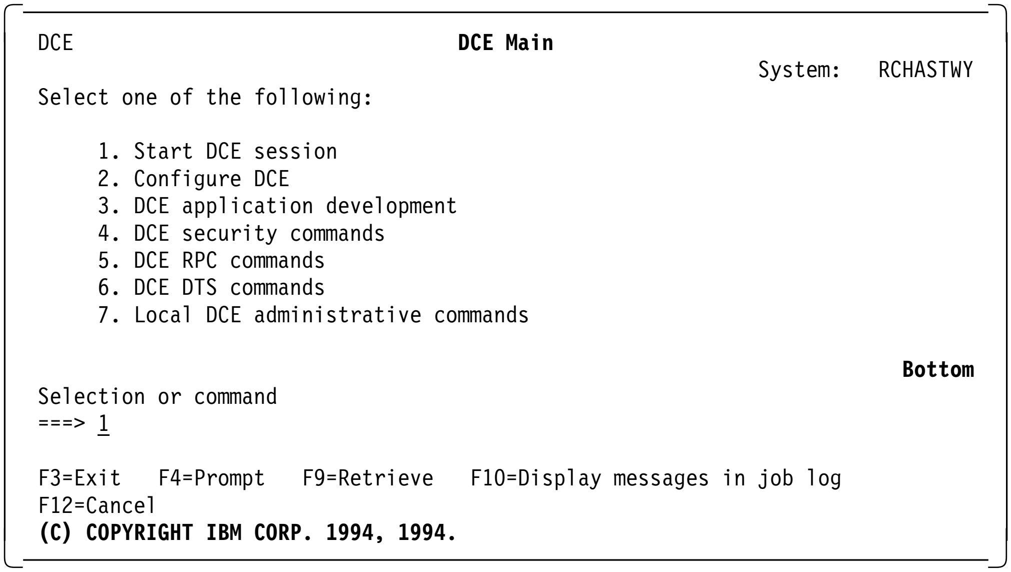
In order to get into OS/400, DCE clients have to contact a DCE server, which is a process listening for service requests. These DCE server processes are started beforehand and run with the security privileges of the OS/400 user(s) who started the server processes. It is up to each DCE server process to check the permissions of the incoming client principal, based on their name, UUID, group memberships, or whatever. If an application implements an ACL manager, the permissions can be administered externally.

So, an incoming DCE client does not have free access to OS/400. The access is restricted to the services a DCE server offers and beyond that, to security and access privileges of the OS/400 user who started the server daemon (listening process).

3.3.4 Administration

DCE Base Services/400 is a comprehensive, integrated set of services that supports the development, use, and maintenance of distributed applications. On the AS/400 system, DCE commands are run by either selecting a command option from a menu or entering a command directly on the command line. DCE Base Services/400 only provides the commands integrated into the OS/400 environment but not the OSF administration commands. In other words, functions, such as rpccp, on the AS/400 will bring up a menu of RPC subcommands that will do many of the same operations found on AIX or OS/2 Warp. The reason that such commands look different on the AS/400 is because the OS/400 command interpreter is so fundamentally different from either AIX or OS/2 Warp. One difference, for example, is that the AS/400 will always fold characters into upper case unless they are enclosed in single quotes.

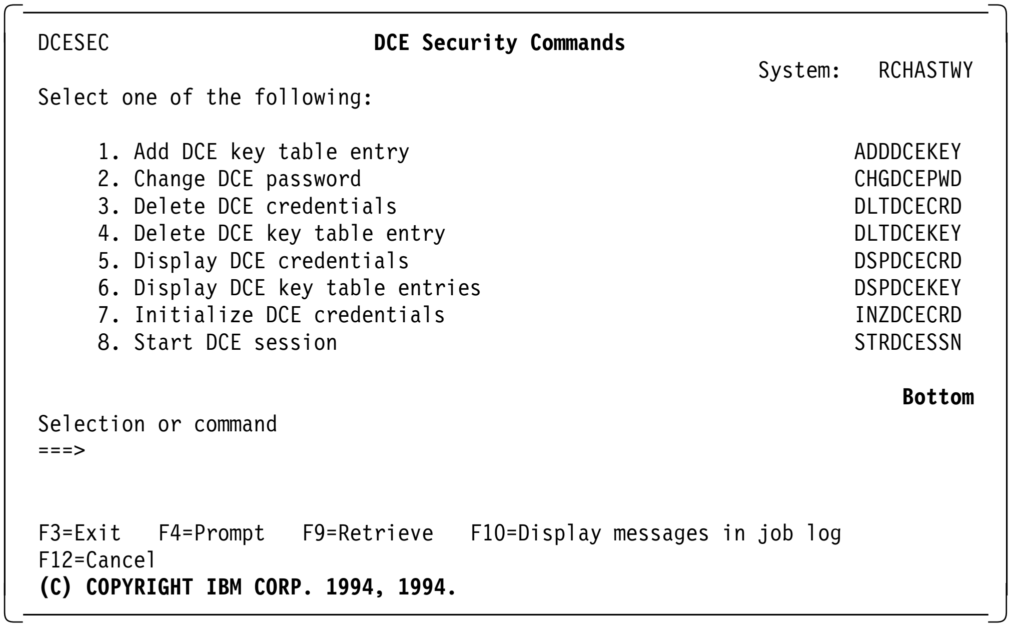
The wrkdce command is used to start the DCE administration suite of menus:



```
DCE                                DCE Main
                                                             System:   RCHASTWY
Select one of the following:

     1. Start DCE session
     2. Configure DCE
     3. DCE application development
     4. DCE security commands
     5. DCE RPC commands
     6. DCE DTS commands
     7. Local DCE administrative commands

                                                                         Bottom
Selection or command
===> 1

F3=Exit   F4=Prompt   F9=Retrieve   F10=Display messages in job log
F12=Cancel
(C) COPYRIGHT IBM CORP. 1994, 1994.
```

Figure 21. DCE/400 Main Screen

Access to security services functions are provided through the following menu (option 4 of the main menu):



```
DCESEC                      DCE Security Commands
                                                              System:   RCHASTWY
Select one of the following:

     1. Add DCE key table entry                                        ADDDCEKEY
     2. Change DCE password                                            CHGDCEPWD
     3. Delete DCE credentials                                         DLTDCECRD
     4. Delete DCE key table entry                                     DLTDCEKEY
     5. Display DCE credentials                                        DSPDCECRD
     6. Display DCE key table entries                                  DSPDCEKEY
     7. Initialize DCE credentials                                     INZDCECRD
     8. Start DCE session                                              STRDCESSN

                                                                          Bottom
Selection or command
===>


F3=Exit   F4=Prompt   F9=Retrieve   F10=Display messages in job log
F12=Cancel
(C) COPYRIGHT IBM CORP. 1994, 1994.
```

Figure 22. DCE/400 Security Commands Screen

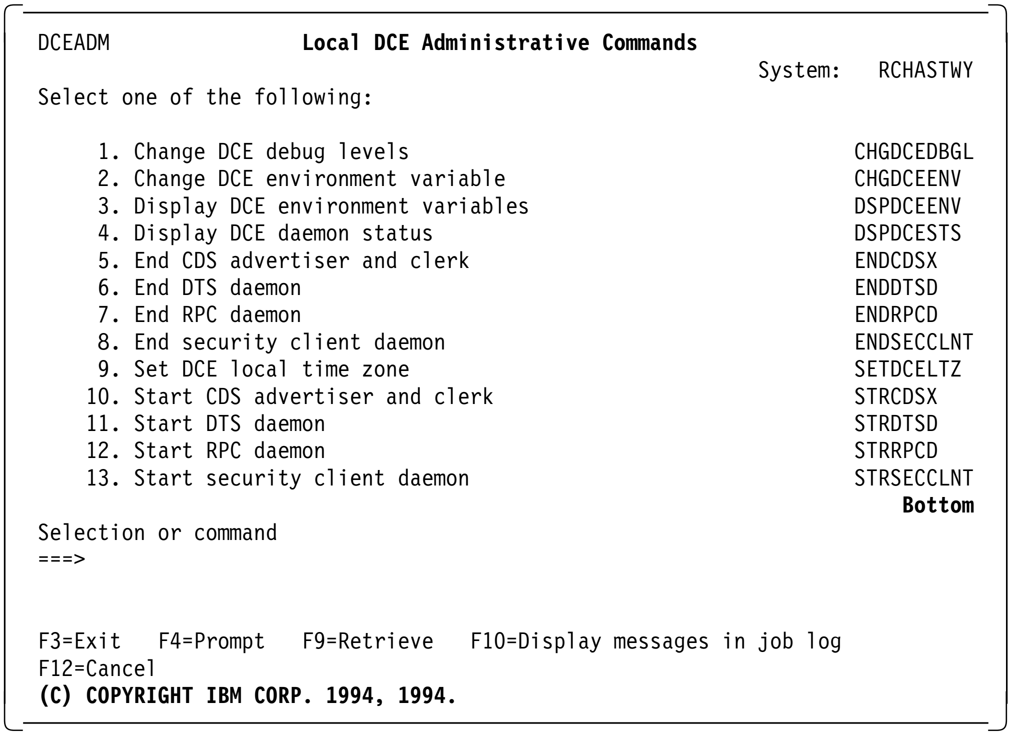
The RPC daemon is a process that provides the Endpoint Map Service, which maintains the local endpoint map for local RPC servers and looks up endpoints for RPC clients. The endpoint, together with the IP address of the server, builds the binding handle to the service, which is kind of a calling address.

There are several commands available to administer RPC bindings, including:

1. Add RPC binding (ADDRPCB) places binding information for an interface identifier.
2. Display RPC entry (DSPRPCE) shows the Name Service Interface (NSI) attributes of a name service entry.

The DTS entity can only be configured as a clerk. The DTS daemon is started with the Start DTS daemon option, which is the STRDTSD command. Also, the administrator has some commands to communicate with the DTS daemon. They are accessible via option 6 of the main menu.

The local administrative commands (option 7) control each individual DCE daemon, such as the RPC daemon or the CDS advertiser and clerk. The daemons can also be started and stopped collectively using the Start Subsystem (STRSBS) and End Subsystem (ENDSBS) AS/400 Control Language commands.



```
DCEADM                 Local DCE Administrative Commands
                                                              System:   RCHASTWY
Select one of the following:

     1. Change DCE debug levels                                        CHGDCEDBGL
     2. Change DCE environment variable                                CHGDCEENV
     3. Display DCE environment variables                              DSPDCEENV
     4. Display DCE daemon status                                      DSPDCESTS
     5. End CDS advertiser and clerk                                   ENDCDSX
     6. End DTS daemon                                                 ENDDTSD
     7. End RPC daemon                                                 ENDRPCD
     8. End security client daemon                                     ENDSECCLNT
     9. Set DCE local time zone                                        SETDCELTZ
    10. Start CDS advertiser and clerk                                 STRCDSX
    11. Start DTS daemon                                               STRDTSD
    12. Start RPC daemon                                               STRRPCD
    13. Start security client daemon                                   STRSECCLNT
                                                                           Bottom
Selection or command
===>


F3=Exit   F4=Prompt   F9=Retrieve   F10=Display messages in job log
F12=Cancel
(C) COPYRIGHT IBM CORP. 1994, 1994.
```

Figure 23. DCE/400 Local Administration Commands Screen

For AS/400 DCE to operate correctly, TCP/IP must be up and running at all times.

3.3.5 Application Development

DCE on OS/400 is able to act as a DCE (RPC) client or server. We give an overview of the tools that are needed to create such applications. Instead of creating a new application from scratch, it is also possible to port either the client or the server part of an existing DCE application to the OS/400, platform. See Chapter 7, “Client/Server Application Development” on page 107 for information on porting applications between platforms.

3.3.5.1 Application Development Tools

To develop DCE applications on the OS/400 platform, only the ILE C/400 compiler can be used to directly call the DCE/400 APIs. Even though you can write applications in other languages, such as ILE COBOL/400, ILE RPG/400 or ILE CL/400, to access the OS/400 data, you have to bind these programs to a *C* program/routine to interact with DCE/400.

To create the UUID (Unique Universal Identifier), run the CRTUUID command with the related parameters:

```
CRTUUID OUTMBR(RLEN) OUTFILE(YOURLIB/IDL) FORMAT(*TEMPLATE)
```

3.3.5.2 Common Programming API Toolkit/400 Feature

Common Programming API Toolkit/400 Feature, which is an optional part of OS/400, provides the necessary environment to DCE. However, because the Common Programming API Toolkit/400 Feature is only usable through ILE C/400, only ILE C/400 can be used to access DCE/400 APIs.

3.3.5.3 IDL Compiler

To create a client/server application, you need to define the interface for your application. This interface consists of RPC routines that handle all the mechanical details of packaging and unpackaging data into messages to send over the network. They also handle sending and receiving of data.

To create this interface, you have to use the IDL (Interface Definition Language) compiler to create the stubs for your client and server applications. This is done with the GENDCESTB (generate DCE stub) command. You have to use C language to write the routines to interface DCE/400.

All the tools available on OS/400 are also available for developing DCE client/server applications. Most of these are provided by the operating system and Integrated Language Environment (C, COBOL, RPG, and Control Language).

3.3.6 Products Packaging and Prerequisites

The DCE Base Services/400 is a separate product. The User Data Privacy feature is an optional, no-charge feature of DCE Base Services/400.

The following table shows the product numbers, depending on the release of OS/400:

Product	OS/400 V3R2	OS/400 V3R7
Operating System (OS/400)	5763-SS1	5716-SS1
DCE Base Services/400	5733-167	5798-TBF
CPA Toolkit, OS/400 Feature	#1690	#1690
ILE C/400	5763-CX2	5716-CX2

Table 6. Ordering Information for DCE on OS/400

3.3.6.1 Hardware Requirements

DCE Base Services/400 operates on AS/400 systems that support OS/400. These processors must operate at a relative performance ratio equal to or greater than a model F02 and be equipped with:

- 8 MB of main storage for operating
- 31 MB of auxiliary storage for configuring
- 31 MB of auxiliary storage for operating
- One tape or cartridge drive for installing DCE/400
- One or more terminals for maintenance and administration functions
- IEEE 802 address is required for installation. There are two options available:
    1. Run the SETIEEE802 command to specify the hardware address of the OS/400 host system. This utility is run outside of the configuration program and thus allows the setting up of the configuration file without running the configuration program
    2. Run the configuration program. When the configuration program detects that there is no LAN card on the OS/400 host system and that no

configuration file exists, a message will be displayed, prompting you for the hardware address

In either case, the configuration file is ignored if a LAN adapter card with a hardware address is installed on the host system.

3.3.6.2 Software Requirements

To operate the DCE Base Services/400 product requires the following:

- Operating System/400, which includes IFS
- Common Programming API Toolkit/400 Feature (feature #1690), which is an optional, free feature of V3R1M0 OS/400
- TCP/IP Connectivity Utilities/400, which is included in OS/400
- Integrated Language Environment C/400 is required for customers who wish to develop DCE applications on the AS/400

3.3.6.3 Installation Tips

Before configuring DCE/400, make sure that the following prerequisites have been satisfied:

- DCE/400 and all its software prerequisites must have been installed on the host system.
  - TCP/IP must be running and the OS/400 host system can communicate with the remote host or hosts on which the DCE Security Server, CDS Server and DTS Server are running. You can use the Verify TCP/IP Connection, VFYTCPCNN command, to ensure that the communication path is set up correctly.
  - The Common Program APIs Toolkit/400 product must be installed on the host system. The QCPA must be included in the library list.
  - The QSYSINC library must be installed on the host system and included in the library list.
- The QDCEADM user profile must exist and be enabled on the OS/400 host. This profile is created during the installation of DCE/400.
- The Security and CDS daemons have already been configured and are running in the cell. At least one DTS server must have been configured and be running in the cell. If an initial cell configuration has been performed, these conditions would have been satisfied.
- The system time on the AS/400 and the DTS server must be within five minutes before starting configuration.
- A supported EBCDIC code page must be set for the program job′s CCSID.

3.4 AIX Implementation of DCE

The IBM RISC System/6000 is one of the reference platforms for DCE. This means that native OSF DCE releases are working on it. It is the first platform on which a fully commercial DCE implementation has been made available.

The IBM RISC System/6000 runs the AIX operating system. The current versions of AIX are Versions 4.1.5 and 4.2 (simply referred to as “Version 4” in the following discussions), and this section will focus on these versions and the DCE functionality provided on these versions. The DCE product name for the AIX platform is *IBM Directory and Security Server for AIX, Version 4*.

The current DCE implementation on AIX is based on OSF DCE 1.1. It provides full client and server support for the entire set of services.

3.4.1 AIX Environment

AIX is a UNIX environment, greatly improved by IBM to allow easy administration and to meet production requirements. This chapter describes the functionality found in AIX Version 4.2. It is the latest version in the family of AIX Version 4. The main characteristics, which are common to most UNIX-like operating systems, are:

- Based on open standards, AIX Version 4 is a robust UNIX operating system. In addition to XPG4 branding, AIX Version 4 complies with the Portable Operating System Interface for Computer Environments (POSIX) IEEE 1003.1-1990 and with IBM′s Open Blueprint client/server computing model. IBM is currently seeking X/Open UNIX95 branding for AIX Version 4.
- A kernel dealing with hardware and hiding it by exposing an interface (system call interface) to commands and applications programs.
- Support for multiple and concurrent users.
- The capability for the operating system and each user to run multiple concurrent tasks (multitasking). Each task is a running program, named a process. Processes running without interruption and not attached to a terminal are called daemons. Processes have their own virtual memory and cannot access other processes′ memory, unless they use shared memory. They can communicate with each other by using many operating system facilities, such as:

  – Signals
  – Semaphores
  – Streams of bytes (using pipes or sockets)
  – Shared memory and others

  Processes are scheduled by the kernel, which tries to allocate CPU resources among them according to priority and previous usage.
- Threads are implemented in the kernel of AIX.
- A byte-stream file format, also called flat files. All files are seen as a sequence of bytes, with no operating-system-enforced format or records. Even binary programs are seen this way.
- A line-oriented interface, known as the Shell, which is used to launch commands or navigate through the file system. Several dialects of the shell are available, but the most common now is the Korn Shell or ksh.
- The support of an optional graphical interface: X-Windows. It allows a program to display its graphical output on a local or a remote machine if needed.
- The Common Desktop Environment (CDE) makes AIX easier to use and helps keep a lid on new training costs.
- A strong TCP/IP implementation with a rich set of functionality.

Some of the most important improvements in AIX over other UNIX implementations are:

- A graphical distributed management tool called SMIT/DSMIT that allows most of the administrative tasks to be performed easily.
- Online documentation with hypertext access through the text or through a graphical tool called InfoExplorer.
- A log-based, crash-proof file system called JFS (Journaled File System).
- Advanced installation tools:
  - Installation Assistant, a Visual System Manager (VSM) tool, uses menus to guide users, even novice users, through the process.
  - EZ-Install, another VSM tool, installs packages of selected software (applications and their prerequisites) based on the type of user chosen for the system; such as general user, system administrator, or application developer.
  - Network Installation Manager (NIM) installs software from a central site to multiple remote clients, virtually eliminating the need for end users to deal with software installation.
- The support of a high-availability solution with the *High Availability Cluster Multi-Processing* (HACMP) product that allows DCE servers to be automatically migrated to another processor in case of a failure and therefore be highly available. The main advantage compared to normal DCE replication is that the migrated server retains its server role (for instance master) in case of failure. So instead of having read-only replicas running, the cell continues to be fully functional, allowing new applications to register and normal administration to take place.
- Support for DCE on the IBM *Scalable Powerparallel System* SP2. The SP2 is controlled by the PSSP control program that uses its own Kerberos authentication system. However, PSSP will not interfere with DCE and thus DCE will run on SP2 nodes just as on any other AIX system. This is also true for the control workstation. From the system planning viewpoint, SP2 nodes, as regards DCE, can be considered just as single AIX systems. From the implementation viewpoint, there are a few things to consider:
  - SP2 nodes are usually multihomed, i.e. they have more than one network interface. If DCE is to be used on certain interfaces only, other interfaces should be explicitly disabled for DCE by means of the RPC_UNSUPPORTED_NETIFS environment variable.
  - SP2 nodes normally do not have a battery-powered system clock and thus lose the actual time whenever powered off. Upon start-up, the clocks are set accordingly to the control workstation's clock. This mechanism may interfere with DCE cell-wide time and the DTS synchronization process.

3.4.2 DCE Implementation Specifics

The DCE implementation on AIX Version 4 is DCE for AIX Version 2.1. It is based on OSF DCE Version 1.1. The highlights are:

- A high performance implementation based upon DCE Version 1.1 supporting advanced levels of security, management, administration, and programming interfaces.
- Provides a comprehensive, integrated set of services for security, naming, interprocess communication, and resource management in a multivendor distributed environment.

- Offers a ″Getting Started″ developers package to enable design and development of secure distributed applications that match business needs.
- Is packaged for convenience; DCE Base Services including all client services are shipped and integrated with AIX Version 4.
- Exploits AIX Version 4 for integrated login and threaded support of powerful multiprocessor systems.
- Supports global data sharing through the base Distributed File System (DFS) available in AIX Version 4.X. Advanced features, such as data replication and online backup, are provided in the Enhanced Distributed File System (EDFS).

The DCE implementation on AIX provides the following features and benefits:

- **Remote Procedure Call (RPC)** Extends the familiar procedure call model by supporting direct calls to procedures on remote systems, enabling programmers to develop distributed applications as easily as traditional, single-system programs. Includes a pipe mechanism to allow bulk transfer of data. Handles data representation conversions transparently.
- **Time Service** Provides fault-tolerant clock synchronization across WANs and LANs for the sequencing, duration, and scheduling of applications.
- **Cell Directory Service** Gives you a common naming service for all resources on the network, such as servers, files, disks, or print queues. Allows users to refer to a resource by the same name even if the network address changes. Greatly increases the efficiency and speed of previous lookups by automatically caching the results of lookups and by replicating services.
- **Global Directory Agent** Provides access to global directory based on industry standards, CCITT X.500/ISO.
- **Security Service** Supports IETF standard GSS-API library. Provides trusted, third-party authentication, registry services, including Extended Registry Attributes (ERA), and access protection through access control lists (ACLs). Provides audit and password strength functionality. Includes security replication features for improved availability.
- **pthreads** Allow you to build applications that can execute multiple tasks within a single process. Based on POSIX 1003.4a Draft 4 standard, mapped to AIX Version 4.1 native threads.
- **Enhanced Distributed File System** Builds on the fundamental features of DCE: security, threads, RPC, directory, and time services. Allows transparent access to files anywhere on the network, including AIX CD-ROM file systems. Improves administration through file sets. Boosts reliability through a log-based physical file system and through the data replication feature.
- **NFS to DFS Gateway** Provides authenticated access from any Network File System (NFS) client to data stored in the DFS file space.
- **Administrative Tools** Allow any machine in the network to perform as an administrative console for such tasks as adding or deleting users, managing namespace entries, and manipulating ACLs on DCE objects. Provide advanced monitoring of DCE environment through optional DCE Manager for AIX product.

3.4.2.1 Core DCE Implementation

DCE is implemented as a set of daemons running with root authority on top of TCP/IP. It does not modify the kernel or any commands.

Each DCE machine has a DCE identity whose principal name is `host/<machine name>/self`. The AIX administrator account *root* automatically inherits the machine principal's credentials.

The daemons running on a client machine are:

- `dced`, the daemon that takes care of RPC client and server programs and the RPC end point mapper. `dced` also handles security validation services.
- `cdsadv`, which is the CDS advertiser daemon that handles access to CDS servers and starts any needed CDS clerks.
- `cdsclerk`, which is the CDS clerk daemon. The CDS clerk handles requests from the client application to a server and caches the results.
- `dtsd`, which is the Time Service daemon, continuously synchronizes the local system clock with the cell's time. The Time Service client daemon is combined with the Time Service server daemon, if used.
- `gdad`, the Global Directory Agent (GDA), if configured.

A DCE server machine is a machine that provides at least one of the DCE services, such as the Security Service, by running the associated server. It can be a master server or a replica server. These machines are normal members of the cell and share the characteristics of a client machine as well. The servers an AIX machine can run are:

- `secd`, which is the security server. It implements authentication of principals and registry of principals' characteristics, including privileges and account information.
- `cdsd`, which is the directory server. It stores and maintains object names within a cell and handles requests to create, modify, and look up data. It needs `cdsadv` to run in order to send advertisements to client machines. If a GDS or DNS is configured, the CDS will communicate with it using the `gdad` and the `gda_child` processes.
- `dtsd`, which is the time server. This is the same process as the client but configured as a server.
- DCE for AIX does not include a DCE Global Directory Service (X.500 Directory server).

3.4.2.2 DFS Implementation

The DCE Distributed File System (DFS) Version 2.1 for AIX is part of an integrated set of distributed services that help deliver secure and robust applications, tools, and services across a network.

Users access and interact with DFS files in the same way they would their local files. All DFS files are part of a single global namespace. Users can access data anywhere in the network without knowing its physical location; they only have to know its name.

The data can actually reside in the DFS native file system, called the Local File System (LFS), in a standard AIX Journaled File System (JFS), or in an AIX CD-ROM file system. The DFS server takes data stored by the physical file system and makes it available to clients through its file exporter. NFS clients can access unprotected DFS data or secured DFS data if they use the AIX NFS/DFS Authenticating Gateway.

Each DFS client in the network accesses files through its cache manager, either in local memory or disk, where it is available for subsequent use. The benefits are that the user has improved response time, and the server machine has less work. Network traffic is also kept to a minimum.

DFS maintains cache coherency among clients, ensuring data integrity. All of this is done by the DFS client code which is—together with DCE core client services—part of the AIX operating system.

The base Distributed File System (DFS) provides the following functionality:

- Provides enterprise file sharing combined with the power of the RS/6000 family of workstations and servers. It ships with AIX Version 4 for convenient connection.
- Ensures reliable and concurrent access to data in both local and remote network environments. Provides a single view of the data from everywhere in the enterprise. Reduces network traffic and file server loading through client data caching. Supports large (>2 GB) filesets and aggregates of data.

IBM provides an enhancement to DFS. It is called Enhanced DFS (EDFS), and it provides the following functionality:

- The enhanced features of EDFS add additional levels of scalability and manageability to base DFS.
- EDFS also ensures higher availability through features such as the LFS, which is a robust, log-based physical file system, and data replication across EDFS servers, which distributes server loads.
- Smart data organization and tools. Administrative tasks are greatly simplified through the EDFS data organization and tools. The concept of data being contained in ″filesets″ provides a convenient way to group administrative tasks, such as data backup, relocation, and replication. The EDFS filesets are location-independent; so they can be transparently replicated or moved to another aggregate for load balancing.

IBM also provides a Network File System (NFS) to the DFS Authenticating Gateway. It provides secure access for legacy NFS clients to data stored in the DFS servers.

The Enhanced DFS Server and the NFS to DFS Authenticating Gateway are not included in the AIX DCE Base Services product. Please refer to Table 7 on page 67 for further details.

3.4.2.3 User Data Masking Encryption Facility for AIX

The DCE security implementation in AIX includes a built-in Data Encryption Standard (DES) encryption facility. This facility is used by the internal authentication mechanisms in DCE.

DES-based versions for user data encryption are available outside the USA and Canada only upon special request and are subject to U.S. Department of Commerce regulations.

An alternative is to use the DCE User Data Masking Encryption Facility for AIX product that provides support for encryption of user data.

Please see Table 7 on page 67.

3.4.3 User Environment

3.4.3.1 User Login Integration

AIX allows for defining an authentication method for the users. The standard method is SYSTEM, which means the password entered by the user is encrypted and checked against the encrypted password stored in the local password file or in the the NIS password map.

An alternative method is to use the Integrated Login Facility in AIX Version 4. This method checks the user ID and the password against the DCE registry and creates the AIX context (such as UID, groups, home directory, and so on) from information stored in the DCE registry. The user does not need to have a local AIX account. The benefit for the user is single login to DCE and AIX. The administrator benefits as well since he/she only has to maintain users in the DCE registry.

By configuring the AIX system for Integrated Login, the AIX login program will use the DCE Security facility to check and validate the user's login, and use the information in the DCE registry to create the login process for the user. From a user point of view, the login is not different from a normal AIX login, but the users will now get a UNIX login environment and at the same time get a DCE login environment, ready to access DFS data and run DCE client and server applications.

Tools are provided with the DCE for AIX products to migrate AIX accounts into DCE accounts and vice versa.

3.4.3.2 Documentation

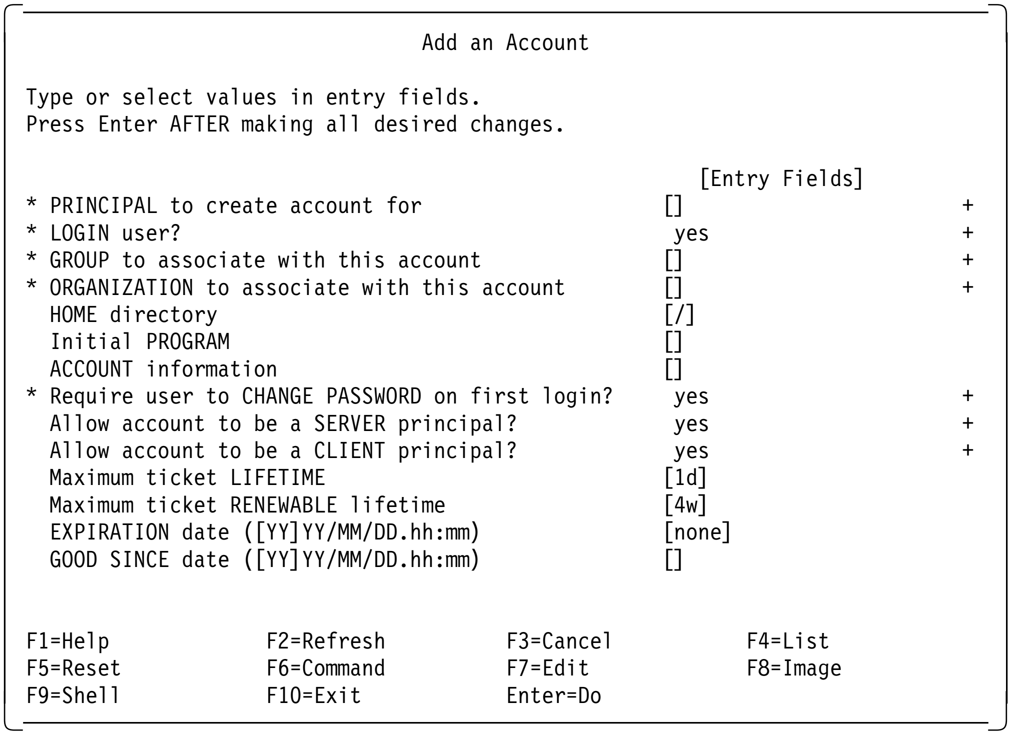
The IBM DCE for AIX documentation is available online in hypertext format with the Information Presentation Facility (IPF) under AIX. This includes an introduction, concept definitions, administrative tasks descriptions, and programming reference manuals.

3.4.4 Administration Tools

AIX DCE 2.1 provides enhanced tools that allow one to configure a simple DCE cell in a matter of minutes. All components are IBM Licensed Program Products (LPPs), where the DCE core clients ship with AIX. Using AIX standard mechanisms and tools, they are easy to install and administer. Installation and configuration can be made using SMIT screens. They allow you to configure DCE and DFS servers and clients very easily. For instance, configuring a one-machine DCE cell running the Security Server, the Directory Server, and a Time Server is a matter of about ten minutes, with most of this time spent waiting for command execution. Configuring DFS servers requires the same amount of time, and configuring clients′ machines can be done in five minutes.

Expert administrators can use the `mkdce` and `mkdfs` commands instead of SMIT to configure DCE and DFS faster or write their own configuration scripts.

Most of the administration tasks can be done with SMIT. This includes managing users, groups, organizations, accounts, passwords, DCE server options and replication, DFS, ACLs, and part of the application servers. For instance, defining an account for a user can be made by filling the first four fields of this SMIT dialog screen:



```
                                  Add an Account

Type or select values in entry fields.
Press Enter AFTER making all desired changes.

                                                         [Entry Fields]
* PRINCIPAL to create account for                     []                      +
* LOGIN user?                                          yes                    +
* GROUP to associate with this account                []                      +
* ORGANIZATION to associate with this account         []                      +
  HOME directory                                      [/]
  Initial PROGRAM                                     []
  ACCOUNT information                                 []
* Require user to CHANGE PASSWORD on first login?      yes                    +
  Allow account to be a SERVER principal?              yes                    +
  Allow account to be a CLIENT principal?              yes                    +
  Maximum ticket LIFETIME                             [1d]
  Maximum ticket RENEWABLE lifetime                   [4w]
  EXPIRATION date ([YY]YY/MM/DD.hh:mm)                [none]
  GOOD SINCE date ([YY]YY/MM/DD.hh:mm)                []


F1=Help             F2=Refresh          F3=Cancel           F4=List
F5=Reset            F6=Command          F7=Edit             F8=Image
F9=Shell            F10=Exit            Enter=Do
```

Figure 24. AIX DCE SMIT Interface for “Add an Account”

The default DCE administration tools are available. The new dcecp command as well as the old rpccp, rgy_edit, or cdscp commands are still useful for writing portable administration scripts and for some uncommon tasks not available in SMIT.

Another level of monitoring administration on AIX can be achieved through the *IBM Distributed Computing Environment Manager for AIX* (DCE Manager for AIX) product.

The DCE Manager for AIX provides advanced management capabilities for heterogeneous DCE installations. With the DCE Manager on an AIX system, you can view and control your entire DCE installation.

Through the NetView for AIX graphical user interface, the DCE Manager allows an administrator to dynamically view the configuration and status of all hosts, services, and servers in the DCE cells.

Integrated with NetView for AIX, which provides a physical network management view, the DCE Manager integrates a logical submap of DCE servers in the same environment.

The DCE Manager provides automatic and dynamic discovery of all DCE servers in the enterprise, decreased set-up, and ongoing maintenance workload.

The DCE Manager allows collection and analysis of management data as input to performance tuning activities or for automation of activities such as startup or load balancing.

The DCE Manager for AIX provides a graphical user interface to monitor all DCE clients and servers in an environment. The application is an integral part of the NetView for AIX program. NetView for AIX provides the physical level of management and the DCE Manager integrates a submap of the logical view of the DCE clients and servers.

The DCE manager dynamically discovers and places DCE clients into the NetView for AIX topology maps, representing your cell(s). The management tool continuously monitors all DCE servers and gives visual notification of status changes. In addition to this, the DCE Manager provides a facility to collect historical data to aid in performance tuning and load-balancing activities.

The DCE Manager is unique in its ability to provide centralized status information to administrators on the health of their DCE configuration. Systems management is a critical component in the future for the enterprise.

Additional information on the IBM DCE Manager for AIX can be found in 8.3, “IBM DCE Manager for AIX” on page 131.

The IBM DCE Manager for AIX product is a separately orderable product. For ordering information, please refer to Table 7 on page 67. Additional information on this product can be found in 8.3, “IBM DCE Manager for AIX” on page 131.

3.4.5 Application Development

Initially, DCE has been defined with C as a primary target language. Other languages can also be used, but not always as easily as C. In both cases, the main issue for an already-trained UNIX programmer is the use of threads.

3.4.5.1 Application Development in C

DCE for AIX comes with development tools needed to program DCE applications. These are:

- The UUIDs generator, uuidgen. UUIDs are needed when defining a new RPC interface or for creating objects managed by application servers.
- The IDL compiler, idl. It converts the interface description into C code that can then be linked with the client and the server to make the remote procedure call transparent.
- The thread library.
- The DCE libraries and include files.
- A set of configuration parameters for the AIX standard C compiler. By invoking the compiler, using the name cc_r4 or xlc_r4, it will use the options and the library needed to compile and link a multithreaded DCE program.
- Thread-aware versions of the AIX standard debuggers dbx and xde. This allows debugging multithreaded programs.

RPC provides a common procedural paradigm, and IDL syntax is like C. The traditional C/UNIX programmer will have to be careful of writing thread-safe code. This is very important when writing a server because a client can usually avoid using threads. You must be aware that most of the AIX libraries are not thread-safe. For instance, this is the case for X-Windows or SNA. The solutions are:

- Write your program as two different processes: one with the DCE part and the other with the non-threaded part. Use an interprocess communication mechanism.
- Use locks to isolate calls to non-reentrant API calls.

3.4.5.2 Application Development with Other Languages

Using other languages for programming, a full DCE application may not be as easy to work with as C. The lack of an appropriate IDL and header files requires extra work. But it is possible to use another language in some cases:

- It is possible to write the DCE part of the application in C and use another language for the functional part. This implies that you write the DCE initialization in C and eventually have some RPC arguments conversion. For a real application, this will be a small part of the code. It is then possible to link the two languages, but you′ll have to be careful because problems due to threads may occur. The solutions are the same when using C with a non-thread-safe library.
- When using C++, the compatibility with C makes it simple to write DCE applications. The only point is to avoid using C++ in the DCE parts.
- When using IBM COBOL POWERbench Version 1 Release 2, DCE support is included with a Cobol IDL and transparent use of DCE services.

IBM provides an "IBM Getting Started with DCE for Application Developers" product that offers DCE code examples, tools, help, and information. Please see Table 7 on page 67.

3.4.5.3 Application Development Tools

Since UNIX was the first platform to provide DCE, it is also the platform on which many third-party development tools are available as products or even as public domain software, such as:

- Higher-level libraries
- Wrapper libraries, procedural or object-oriented
- Generators that are able to create, for instance, IDL files and part of the server code

Their common objective is to reduce or even completely hide the complexity of the low-level, but very powerful, standard DCE APIs. See Chapter 7, "Client/Server Application Development" on page 107 for more information about such products.

3.4.6 Planning Considerations

RISC System/6000 machines are a good solution for most of the DCE applications. They are the preferred platform for the following machine roles:

- As DCE core servers, such as the Security Server, Directory Server, Time Server, and DFS servers. When planning to use an AIX machine for this purpose, you must be careful that they may be CPU and memory consuming, depending on the size of your cell and its activity. Because of the good CPU performance/price ratio of the RISC System/6000, its capability to manage large amounts of memory and the availability of a full and stable version of

DCE with HACMP support, it is a platform of choice to be any kind of server in the cell.

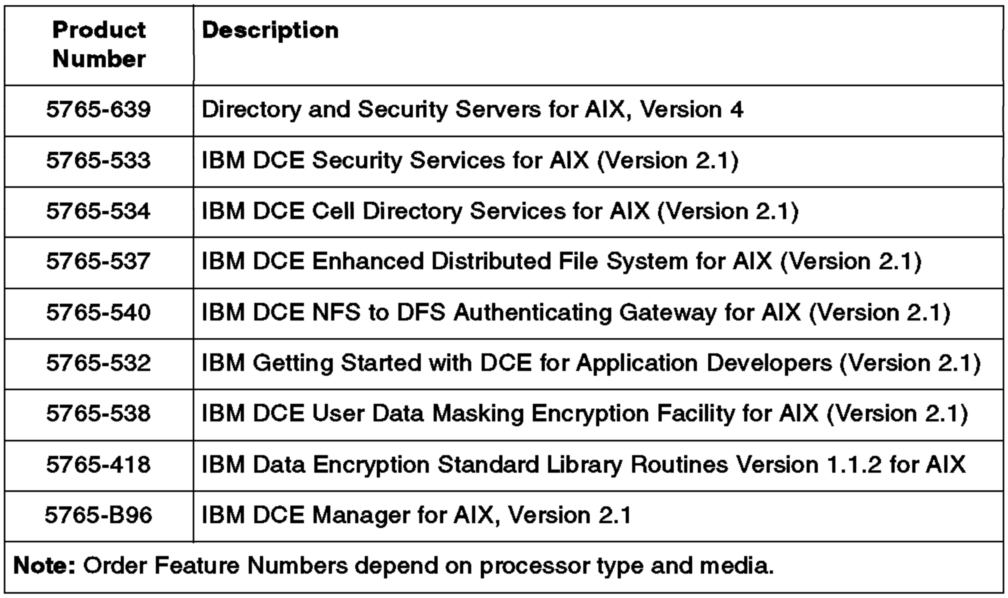
- As DCE application servers for the following types of applications:
  - AIX programs, for instance, scientific number-cruncher applications.
  - Interfaces to other AIX programs, for instance, sending requests to accounting software through a proprietary non-DCE API.
  - Interfaces to other non-AIX programs, for instance, sending requests to MVS/IMS through a LU 6.2.
  - Transaction monitors and application servers in three-tiered client/server models running CICS/6000 or Encina.

  Or they can be whatever you imagine. DCE application servers interfacing to an existing CICS or IMS application would be implemented preferably on a mainframe with DCE for OS/390.

- As DCE clients. Low-end RISC System/6000s are the preferred platforms for DCE clients, which usually run graphical applications on user's desks.

3.4.7 Product Packaging and Prerequisites

The following table gives an overview on order numbers of the different DCE products available for AIX:



Product Number	Description
5765-639	Directory and Security Servers for AIX, Version 4
5765-533	IBM DCE Security Services for AIX (Version 2.1)
5765-534	IBM DCE Cell Directory Services for AIX (Version 2.1)
5765-537	IBM DCE Enhanced Distributed File System for AIX (Version 2.1)
5765-540	IBM DCE NFS to DFS Authenticating Gateway for AIX (Version 2.1)
5765-532	IBM Getting Started with DCE for Application Developers (Version 2.1)
5765-538	IBM DCE User Data Masking Encryption Facility for AIX (Version 2.1)
5765-418	IBM Data Encryption Standard Library Routines Version 1.1.2 for AIX
5765-B96	IBM DCE Manager for AIX, Version 2.1
Note: Order Feature Numbers depend on processor type and media.	

Table 7. IBM DSS and DCE Products for AIX

DCE core client services, including the DFS client, are packaged with AIX Version 4. The Server Services can be ordered separately as single products, for example *IBM DCE Security Services for AIX (Version 2.1)*, or the two main Server Services, Security and Directory Server, can be ordered together in a single package, called *Directory and Security Server for AIX, Version 4* (DSS for AIX). Although the version numbers differ, they are basically the same code.

The products also include all documentation in IPF format (Information Presentation Facility, also included with the products) and some also in PostScript format.

DSS for AIX Version 4 (or AIX DCE 2.1) can be installed on any RISC System/6000 running AIX Version 4. It requires TCP/IP be installed and configured on top of an Ethernet or Token-Ring interface to provide a hardware address, but it can then communicate using any TCP/IP-supported media, such as SLIP or X.25.

Installation is made with SMIT, the System Management Interface Tool, from the Software Installation & Maintenance menu. It is possible to choose which parts to install from a list. Be aware that you may need Program Temporary Fixes (PTFs) to be installed on your system to install DCE. If they are on the same media as DCE, `installp` will automatically install them. If not, you will have to ask for them from your IBM support.

3.5 OS/2 Warp Implementation of DCE

The current DSS implementation on OS/2 Warp is based on OSF DCE Version 1.1. The full product names for the server and the client parts are: *IBM Directory and Security Server for OS/2 Warp, Version 4* **and** *DCE/DFS Client for OS/2 Warp, Version 4*. This DSS implementation is part of the IBM Software Server series. DSS for OS/2 Warp enhances and complements OS/2 LAN Server and OS/2 Warp Server by means of providing DCE services to those products.

DSS for OS/2 Warp supports threads and provides full client and server support for RPC, CDS, Security Service, DTS and DFS client. In addition, it ships with a sophisticated graphical user interface for administration of DSS and any DCE resources.

3.5.1 OS/2 Warp Environment

OS/2 Warp is an operating system running on Intel platforms with a 386 or higher processor. It is a true 32-bit system with multitasking and multithreading capabilities. It provides both a graphical interface, named Presentation Manager, (PM) and a command line-oriented interface.

Due to OS/2 Warp's multitasking capability, a user can run several programs at the same time. Each program is called a task. Sometimes, the UNIX terminology is also used, and a program is called a process, or a daemon, if it runs as a permanent background task in order to provide system-level functions. The sophisticated multitasking and multithreading capabilities of the Warp kernel provide an excellent base for distributed, network-oriented services like DSS.

Like UNIX, OS/2 Warp inherently conforms to all POSIX standards that are relevant for DCE.

For group computing, specifically disk, file, and printer sharing, OS/2 LAN Server and OS/2 Warp Server are the IBM products being used. DSS for OS/2 Warp extends the functionality of these products by adding:

- DCE login and security features for single user sign-on to OS/2 LAN Server or OS/2 Warp Server and to DCE services
- File and printer sharing services across multiple OS/2 LAN Server or OS/2 Warp Server domains
- Consolidated and simplified administration of multiple OS/2 LAN Server or OS/2 Warp Server domains

OS/2 LAN Server and OS/2 Warp Server incorporate the concept of *domains*. A domain is a collection of machines, users, and other resources, administered from a single point by means of a domain controller. Such domains are typically established throughout single departments with a size of 10 to 100 users (although they can be smaller or bigger). From an administrative perspective, each domain is an independently managed unit:

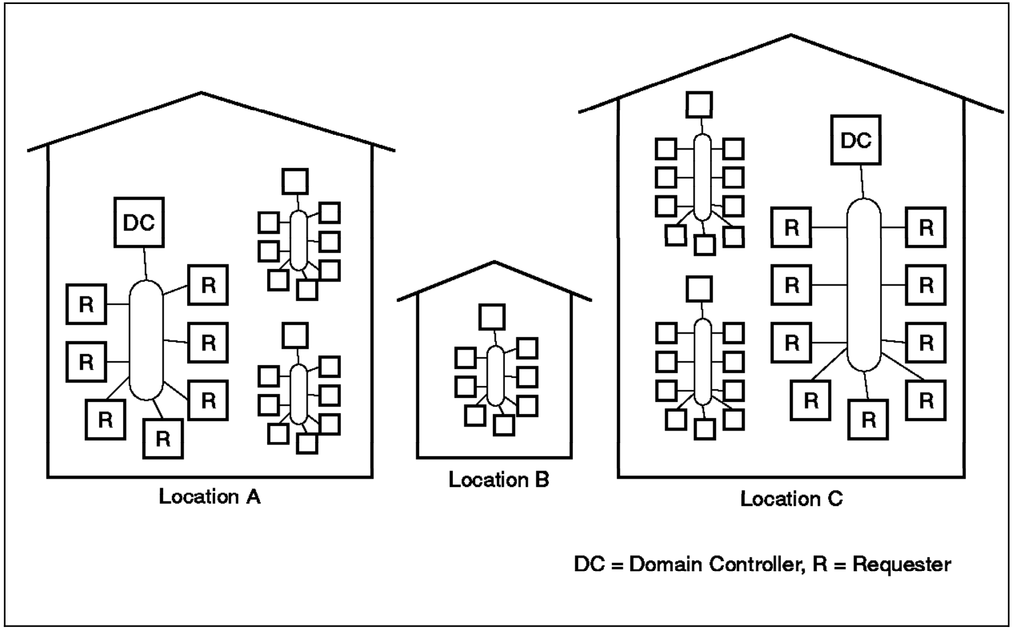


Figure 25. OS/2 LAN Server or OS/2 Warp Server Domains (Example)

Figure 25 shows, very simplified, such a domain layout, having a total of seven domains at three different locations. Since a domain forms a unit of administrative authority, a user would need to have multiple accounts to work in different domains.

DSS for OS/2 Warp now provides the ability to form DSS cells that comprise several OS/2 LAN Server or OS/2 Warp Server domains and thus form a higher level of administrative unit.

In the example illustrated in Figure 26 on page 70, a user (or any other resource) is now defined and administered on a per-cell basis as opposed to on a per-domain basis prior to DSS for OS/2 Warp implementation. Within a DSS cell, any resource is known cell-wide; thus a user in a domain at location A can print on a printer or access a file in location B, provided he has the necessary permissions. To centralize administration even more, all of the LAN domains in Figure 26 on page 70 could have been consolidated into one single DSS cell.

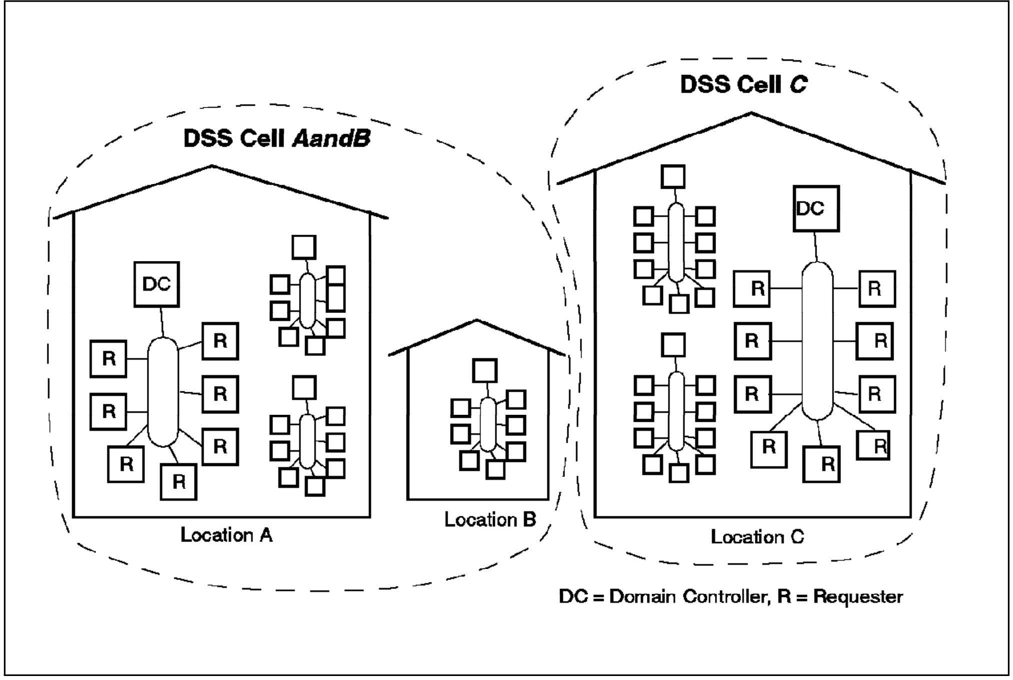


Figure 26. DSS Cells Comprising LAN Server or Warp Server Domains (Example)

Moreover, DSS for OS/2 Warp allows you to introduce a new, hierarchical management structure, called Resource Domains. In the example given in Figure 26, the three LAN Server domains at location A could form such a resource domain, while the single LAN Server domain at location B could form another resource domain. An administrator at location A can then administer all resources through a single user interface, or he/she can, due to the hierarchical nature of DSS administration, allow sub-administrators to manage only parts of the DSS cell.

Directory and Security Server for OS/2 Warp, Version 4, seamlessly integrates the directory and security functiona lity of DCE with an existing OS/2 LAN Server, Version 4.0, or OS/2 Warp Server, Version 4, system. It provides a transparent replacement for the OS/2 LAN Server and the OS/2 Warp Server directory and security functions.

A separately orderable package, &DSS_Warp_Cli., offers DCE and DFS client functions to any OS/2 Warp workstation. A separate DFS server, such as an IBM RISC System/6000 with the appropriate DFS server software, must exist somewhere in the network in order to take advantage of the distributed file system. DCE client functions in this package comprise DTS client, security client and directory client.

3.5.2 DCE Implementation Specifics

DSS for OS/2 Warp is not simply an OSF DCE implementation on the OS/2 Warp platform. Together with OS/2 LAN Server or OS/2 Warp Server, it is an integrated DCE-based service that enhances those products and also offers standard DCE functionality and APIs to other applications. DSS for OS/2 Warp replaces the directory and security services of the OS/2 LAN Server and the OS/2 Warp Server upon installation.

DCE services require threads that conform to the POSIX 1003.4a Draft standard. As OS/2 Warp native threads are different, DCE threads are implemented on top of them by mapping the DCE interface to OS/2 Warp threads.

For non-DCE applications, DSS for OS/2 Warp offers the standardized GSS-API on the OS/2 platform.

The Directory Server part of DSS supports the X/Open Directory Service (XDS) API for direct access, or the RPC Name Service Independent (NSI) interface for indirect access.

DCE is designed to run over the TCP/IP protocol. However, DSS on the OS/2 platform can use sockets over TCP/IP or over NetBIOS to communicate between clients and servers. The selection is made at installation time. The file and print components use NetBIOS over TCP/IP.

The GDA (Global Directory Agent) function is provided by DSS for OS/2 Warp. It supports DNS as a global directory service. This allows OS/2 Warp platforms full integration in a multicell network. The GDS (Global Directory Service) function is not provided by DSS for OS/2 Warp.

3.5.3 User Environment

Directory and Security Server for OS/2 Warp, Version 4, enables the users to log in to the OS/2 LAN Server or OS/2 Warp Server domain and to the DSS cell in a single, integrated login step. In addition to the services provided by the LAN Server or Warp Server, a user in a domain that has been migrated to DSS will also be able to access and use resources in other domains because they are managed all by a single DSS server. Because the user also automatically gets DCE network credentials, he/she may even launch DCE applications in the cell and participate in a multiplatform computing environment.

If the user's machine is also set up as a DFS client, it can participate in the location-independent DFS filespace, and the user can access files from any DFS server, such as an IBM RISC System/6000 with AIX and the appropriate DCE/DFS server software.

Complete online documentation is supplied on the product CD-ROM as Information Presentation Facility (IPF) books, and some are also available as PostScript files for the convenience of easy printing. For more details, see 3.5.7, "Product Packaging and Prerequisites" on page 74.

3.5.4 Administration Tools

IBM DSS for OS/2 Warp, Version 4, can be installed, configured, and used with common OS/2 Warp procedures, including CID.

Directory and Security Server for OS/2 Warp, Version 4, comes with a full graphical administration user interface (GUI) that integrates with the administration interface of the OS/2 LAN Server and OS/2 Warp Server. Users and groups can be managed from within this single GUI, no matter whether they are DSS users or non-DSS users (for example, users and groups that remain to be managed by an OS/2 LAN Server or an OS/2 Warp Server). This GUI allows management of all DSS (DCE) resources in the cell.

The default DCE administration command, dcecp, is also available as a line-oriented command and provides access to all DCE-related configuration and administration services. The net <...> commands from the OS/2 LAN Server or OS/2 Warp Server environment are extended throughout the installation of DSS for OS/2 Warp.

Tools are shipped with the Directory and Security Server for OS/2 Warp, Version 4, that allow you to easily reorga nize existing OS/2 LAN Server or Warp Server domains. They allow you to merge and to move domains as well as to move servers.

For further discussions about administration tools, see Chapter 8, "DCE Administration Tools" on page 129.

3.5.5 Application Development

For application development, DCE has been defined with C as a primary target language. Other languages can also be used, but not always as easily as C.

3.5.5.1 Standard DCE Application Development Toolkit

IBM Directory and Security Server for OS/2 Warp, Version 4 comes with a development toolkit needed to write DCE applications. It includes:

- The UUIDs generator, uuidgen. UUIDs are needed when defining a new RPC interface or for creating objects managed by application servers
- The IDL compiler, idl. It converts the interface description into C code that can then be linked with the client and the server to make the remote procedure call transparent
- The DCE libraries and include files
- Diagnostic Tools
- Examples

RPC provides a common procedural paradigm, and IDL syntax is like C. As OS/2 Warp is already multithreaded, an experienced OS/2 C programmer will familiarize easily with DCE programming. See also Chapter 7, "Client/Server Application Development" on page 107.

3.5.5.2 Application Development with Other Languages

Using other languages for programming a full DCE application is not easy. The lack of an appropriate IDL and header files makes it a huge amount of work. But it is possible to use another language in some cases:

- It is possible to write the DCE part of the application in C and use another language for the functional part. This implies that you write the DCE initialization in C and eventually have some RPC arguments to convert. For a real application, this will be a small part of the code. It is then possible to link the two languages.
- When using C++, the compatibility with C makes it simple to write DCE applications. The only point is to avoid using C++ in the DCE parts.

3.5.5.3 Third-Party Application Development Tools

As with UNIX, some third-party development tools are available as products or even as public domain software, such as:

- Higher-level libraries
- Wrapper libraries, procedural or object-oriented
- Generators that are able to create, for instance, IDL files and part of the server code

Their common objective is to reduce, or even completely hide, the complexity of the low-level, but very powerful, standard DCE APIs. See Chapter 7, "Client/Server Application Development" on page 107 for more information about such products.

3.5.6 Planning Considerations

DSS for OS/2 Warp significantly enhances the functionality and lifts some natural restrictions from the OS/2 LAN Server or OS/2 Warp Server products. Such a natural restriction was the limitation of the scope of management to a single domain. With the addition of DSS for OS/2 Warp, this limitation has gone; multiple OS/2 LAN Server or OS/2 Warp Server domains can be combined together with a single registry and directory database that best fits the actual requirements of a certain organization.

If DSS for OS/2 Warp is being installed in an existing environment, tools are provided with the product that help to reorganize the current domain structure to build up a new, more organization-oriented structure of administrative responsibilities. Current users, groups, and other resources will be migrated to the new DSS databases during installation, or afterwards through the use of provided tools.

OS/2 Warp machines are an excellent and cost-effective desktop platform for running DCE distributed applications. With DSS for OS/2 Warp, distributed applications can span multiple platforms, having for example powerful UNIX systems as departmental database and application servers and OS/2 Warp machines as clients. The common base, DCE and RPC, integrates such different platforms into one common application platform. Simple printer and file sharing is probably still best done within the same operating system platform using OS/2 LAN Server or OS/2 Warp Server. Distributed File System (DFS) then can act as a corporate data repository with a high level of access control facilities.

For best availability, Directory and Security Server for OS/2 Warp, Version 4, supports replication of the Securit y or Directory Servers. Replication allows read-only access to the services even if the main server is unavailable, allowing most applications to continue without interruption since applications most often do not write to the servers.

Replication of DSS/DCE core services should be planned ahead together with the domain and cell layout. Replication not only improves the availability in case of failures, it may also improve performance due to load balancing among multiple servers.

3.5.7 Product Packaging and Prerequisites

The following table gives an overview on order numbers of the different DSS and DCE products available for OS/2 Warp:

Order Type Number	Feature	Part Number	Description
US, Canada:5801-AAR	2035	10H9754	Directory and Security Server for OS/2 Warp, Version 4 (DES)
US, Canada: 5801-AAR	2036	25H7945	Directory and Security Server for OS/2 Warp, Version 4 (CDMF)
Others: 5622-967	5895		
US, Canada: 5801-AAR	2037	25H7940	DCE/DFS Client for OS/2 Warp, Version 4 (DES)
US, Canada: 5801-AAR	2038	25H7946	DCE/DFS Client for OS/2 Warp, Version 4 (CDMF)
Others: 5639-A18	5895		

Table 8. IBM DSS Products for OS/2 Warp (Excerpt)

In addition to the products and numbers listed in Table 8, depending on special terms and conditions, such as authorization to copy licensing, other feature and part numbers may apply. The ordering numbers in Table 8 are for US English versions only. For other languages, please refer to the country-specific announcements. DES-based versions are available outside the USA and Canada only upon special requests and are subject to U.S. Department of Commerce regulations.

The server packages also include the client packages, which therefore do not need to be ordered separately for installation on a server.

IBM Directory and Security Server for OS/2 Warp, Version 4, runs on all OS/2 Warp-capable hardware with at least 24 MB of memory and at least 400 MB hard disk. As a minimum, an Intel processor model 80486 running at 33 MHz is recommended for the servers; 66 MHz would be required if the administrator user interface is being used.

IBM Directory and Security Server for OS/2 Warp, Version 4, requires OS/2 Warp with the latest FixPack applied and either the IBM OS/2 LAN Server, Version 4, or the IBM OS/2 Warp Server, Version 4.

The products ship on CD-ROM media that also contains softcopy versions in IPF and PostScript formats of the following publications:

- *Getting Started*
- *Directory and Security Server Up and Running* (IPF only)
- *Directory and Security Server Administrator's Reference* (IPF only)
- *Directory and Security Server Command and Utilities* (IPF only)
- *Directory and Security Server Client User's Guide* (IPF only)
- *Directory and Security Server Problem Determination* (IPF only)
- *Directory and Security Server Trademarks* (IPF only)
- *Directory and Security Server Programming Guide and Reference* (IPF only)
- *MPTS Configuration Guide* (IPF only)
- *MPTS Sockets Programming Reference* (IPF only)
- *Administration Guide*

- *Administration Commands Reference*
- *Application Development Reference*
- *Application Development Guide — Introduction and Style*
- *Application Development Guide — Core*
- *Application Development Guide — Directory Services*
- *DFS Client Guide and Reference*
- *Error Message Manual* (IPF only)
- *Glossary* (IPF only)

3.6 DOS/Windows Implementation of DCE

The current DCE implementation from IBM on DOS/Windows is based on OSF DCE 1.0.1 and originally developed by Gradient Technologies. It provides full client and server RPC support, threads and client support for CDS, security service, and DTS.

For a description of IBM DCE for Windows NT, see 4.1, “IBM DCE for Windows NT” on page 81.

3.6.1 DOS/Windows Environment

Windows is a graphic interface on top of the popular DOS operating system. It also provides many extensions including:

- A better memory model, allowing 32-bit programs to run.
- A virtual memory model with the capability of using disk swap.
- A non-preemptive multitasking capability. This means that multitasking is not fully enforced by the operating system; so applications must be well-behaved and yield the CPU regularly.
- A standard API to access to TCP/IP layers, called Window Socket or just WinSock, originated from the DLL file, `WINSOCK.DLL`.

In Windows terminology, a running application is called a task. There is no threads implementation within the operating system.

3.6.2 DCE Implementation Specifics

DCE requires threads that conform to the POSIX1003.4a Draft standard. Since Windows does not support threads, by default, a product is needed to provide them. This implementation has several limitations due to Windows and DOS design. For instance, the threads themselves are not preemptive. So, the programmer has to be careful with scheduling. Extensions have been added to let the programmer explicitly yield the CPU.

DCE is implemented as a set of tasks that can be launched automatically at startup, or at any moment, by clicking on icons. It needs a TCP/IP layer, and currently supported products include:

- LAN Workplace for DOS from Novell, Versions 4.01 and V 4.1
- PC/TCP from FTP Software, Versions 2.05, 2.1, and 2.2
- Pathworks TCP/IP from DEC
- TCP from 3Com, Version 2.0
- Any Windows Sockets Version 1.1-compliant transport

You have to be careful that only some versions of the previous TCP/IP layers are supported. When using another version, even the newest one, you may face unpredictable behavior, including system crash. You should also notice that the use of the WINSOCK.DLL interface solves many compatibility problems, but has a strong impact on performance.

A Windows machine is always a DCE core services client machine; it cannot run any DCE core server. It can run application servers. The tasks running on a Windows client machine are:

- RPCD, which provides the endpoint map service. It allows DCE application servers to run.
- CDSCLERK, which is an interface between CDS client applications and CDS servers. The CDS clerk handles requests from client applications to a server and caches the results returned by the server.
- DTSD, which synchronizes the local clock if DTS is used.

The access to DCE is available for Windows applications only. DOS applications, even those running as a Windows task, cannot use the libraries.

3.6.3 User Environment

The installation and configuration tools provide online help, using the standard Windows mechanism. There is no other online documentation.

The user environment is not modified. If users want to get their DCE network credentials, they have to use the line-oriented DCELOGIN command, providing their DCE principal (user) name and password.

The user can then use commands, such as KLIST, KINIT, and KDESTROY, respectively, to see their credentials, change their options and lifetime or destroy it. Because Windows does not provide a line-oriented interface, these commands run in their own windows. They are mainly a revamping of the original, line-oriented commands.

As far as access to and from DCE is concerned, the Windows environment works the same way; it is also a single-user environment.

3.6.4 Administration Tools

IBM DCE for Windows can be installed, configured, and used with usual procedures.

Configuration is made by calling the PCDCECFG program. It allows you to fully configure DCE with a standard Windows interface. You will have to be careful since performance depends on configuration. You should take note of the following points:

- If possible, choose a TCP/IP transport supported by DCE for Windows. Avoid using the WINSOCK interface for performance reasons.
- Use the PE_SITE file to locate the security server instead of going through CDS. This allows a client to connect directly to a security server, which speeds up boot time.
- Avoid using DTSD. Although simple in its function, DTSD uses a complex algorithm to determine and set system clocks. In a DOS/Windows environment, it uses a relatively large amount of memory and CPU

resources. However, system clocks must be synchronized within an acceptable tolerance (DCE default is 5 minutes) with an alternative method when not using DTSD.

The default DCE administration tools, such as RPCCP, CDSCP, RGY_EDIT, ACL_EDIT, and DTSCP are also available. However, because Windows does not provide a line-oriented interface, these commands run in their own windows. They are mainly a revamping of the original, line-oriented commands.

With a few others, these commands allow you to perform most or all the administration of a cell, with the drawback of a primitive interface.

3.6.5 Application Development

DCE has been defined with C as a primary target language. Other languages can be used also, but not always as easily as C.

Moreover, the DCE implementation in the Windows environment has to accommodate with Windows and DOS technical limitations. So, you should be very careful of possible technical problems when using tools, APIs, and languages that are not explicitly known to work with DCE for Windows.

3.6.5.1 Application Development in C

IBM DCE SDK for OS/2 and Windows comes with development tools needed to program DCE applications. These include:

- The UUIDs generator, UUIDGEN. UUIDs are needed when defining a new RPC interface or for creating objects managed by application servers.
- The IDL compiler, IDL. It converts the interface description into C code that can then be linked with the client and the server to make the remote procedure call transparent.
- The DCE libraries and include files.

RPC provides a usual procedural paradigm, and IDL syntax is like C. So, the experienced C programmer will mainly have to be careful of writing thread-safe code. This is very important when writing a server because clients can usually avoid using threads.

Microsoft C/C++ and the IBM DCE SDK for OS/2 and Windows are the usual tools to do Windows DCE application development.

3.6.5.2 Application Development with Other Languages

Using other languages for programming a full DCE application is not easy. The lack of an appropriate IDL and header files make it a huge amount of work. But it is possible to use another language in some cases:

- It is possible to write the DCE part of the application in C and use another language for the functional part. This implies that you write the DCE initialization in C and eventually have some RPC arguments to convert. For a real application, this will be a small part of the code. It is then possible to link the two languages.
- When using C++, the compatibility with C makes it simple to write DCE applications. The only point is to avoid using C++ in the DCE parts.

3.6.5.3 Application Development Tools

There are many DCE application development support products on the market, mostly wrapper libraries, that allow such popular Windows development tools as PowerBuilder or Visual Basic to interface with DCE. See Chapter 7, "Client/Server Application Development" on page 107 for more information about such products.

3.6.6 Planning Considerations

The ability to use popular GUI-builder tools for DCE application development makes a Windows workstation an attractive DCE client.

Because of the technical limitations of Windows, it is not a good idea to use it to run DCE application servers. This would introduce not only a lot of programming effort, but also may lead to performance problems.

3.6.7 Product Packaging and Prerequisites

The following table gives an overview on order numbers of the different DCE products available for Windows:

Order Type Number	Feature Number	Part Number	Description
5696-657		59G5689	IBM DCE SDK for OS/2 and Windows Version 1.0
5871-AAA	6199	96F8690	IBM DCE SDK for OS/2 and Windows Version 1.0 with User Data Privacy
5696-690		59G5662	IBM DCE Client for Windows Version 1.0
5871-AAA	6201	96F8692	IBM DCE Client for Windows Version 1.0 with User Data Privacy

Table 9. IBM DCE Products for DOS/Windows

IBM DCE for Windows runs on any Intel platform with a 386 processor or higher, at least 4 MB of memory, 5 MB of free disk space, and 5 MB of Windows swapfile. It requires DOS 5.0 or higher and Windows 3.0 or 3.1.

Installation can be made from diskettes or hard disk. It is done with the usual INSTALL program, with a graphical interface. The installation program creates directories, installs the appropriate files, and creates the initial configuration, including modifications in the CONFIG.SYS file and setting necessary environment variables.

3.7 IBM DCE Cross Platform Matrix 11/96

The following table summarizes the availability of DCE functions on IBM platforms. For DCE implementations on other platforms, see the following chapter.

DCE Function	DOS/ Windows (Win NT *)	OS/2 Warp	AIX/6000	OS/400	VM/ESA	OS/390
OSF Level	1.0.1	1.1	1.1	1.0.2	1.0.3	1.1
DCE Core Client Functions						
Threads	✓	✓	✓	✓	✓	✓
RPC Client	✓	✓	✓	✓	✓	✓
Directory Service (CDS) Client	✓	✓	✓	✓	✓	✓
Time Service (DTS) Clerk	✓	✓	✓	✓	Sub	✓
Security Service Client	✓	✓	✓	✓	✓	✓
DCE Core Server Functions						
RPC Server	✓	✓	✓	✓	✓	✓
Time Service (DTS) Server		✓	✓			✓
Directory Service (CDS) Server		✓	✓	SoD		SoD
Security Server		✓	✓	SoD		✓
X.500 Inter-Cell Access						
Global Directory Service (Client)		✓	✓			
Global Directory Service (Server)						
Data Sharing Services						
Distributed File System Client	ISV	✓	✓	SoD		✓
Enhanced DFS Server			✓			✓
Other Services						
Encina Clients	ISV	✓	✓			
Encina Servers			✓			
Application Support IMS and CICS	N/A	N/A	N/A	N/A	N/A	✓

Notes:

✓ Implemented today
ISV Available from an Independent Software Vendor
SoD Statement of Direction
N/A This feature is not applicable to this platform
Sub A substitute function exists for coarse time correction
(*) For Windows NT see Chapter 5, “DCE Implementations on Non-IBM Platforms”

Table 10. IBM DCE Product/Function Matrix (11/96)

Note: For Windows NT, see 4.1, “IBM DCE for Windows NT” on page 81.

Chapter 4. DCE Implementations on Non-IBM Platforms

DCE implementations are also available on several non-IBM platforms and operating systems. Most often, it is implemented by the hardware/software vendor itself or by independent software vendors. This chapter gives an overview of such products, providing information as it can be found from public sources, such as the World Wide Web. However, it is not possible to cover all available products from any vendor.

4.1 IBM DCE for Windows NT

IBM Distributed Computing Environment for Windows NT introduces an IBM product for the Windows NT platform. It provides a runtime environment for NT clients as well as security and cell directory servers and sophisticated GUI-based administration tools. For customers who want to integrate OS/2 Warp systems into their Windows NT environment, a DCE and DFS client package for OS/2 Warp is also included in IBM DCE for Windows NT.

IBM DCE for Windows NT is based on Digital Equipment's Implementation for Windows NT, except for the DCE and DFS client for OS/2 Warp, which is the same as that included in IBM DSS for OS/2 Warp (see 3.5, "OS/2 Warp Implementation of DCE" on page 68).

4.1.1 Components

IBM DCE for Windows NT consists of the following five components:

1. **DCE Runtime Services for Windows NT**

   The runtime services are needed at least for DCE clients and provide support for threads, time service, RPC, security, and cell directory clients. Threads are integrated in Windows NT's native treads.

   The runtime services also includes Windows-based DCE management tools:

   - *DCEsetup* for configuring and managing the services on a system.
   - *DCE Director* for managing DCE cells. It allows you to create, change, and delete DCE opjects, such as users, groups, hosts, CDS directories, and servers. It also allows you to access DCE's native administration commands (`rgy_edit`, `cdscp`, `acl_edit`, and `dtscp`) and adds new functions like preconfiguration of hosts.
   - *Visual ACL* is an Windows-based ACL editor for graphically managing DCE ACLs.

2. **DCE Cell Directory Services for Windows NT**

   The cell directory server for Windows NT includes a Global Directory Agent (GDA), which provides a means of linking multiple CDS namespaces via either X.500 or the Internet Domain Name Service (DNS).

3. **DCE Security Services for Windows NT**

   This is a standard OSF DCE implementation. It does, however, not support data encryption.

4. **DCE Application Developer's Kit for Windows NT**

Contains standard tools for developing DCE applications, such as an IDL compiler, samples, libraries, and header files. IDL support is provided from OSF DCE 1.2, which includes C++ language and support for distributed objects.

5. **DCE Client Including DFS for OS/2 Warp**

   DCE and DFS client code for OS/2 Warp is included for customers to easily incorporate OS/2 Warp systems in their Windows NT environment. DFS servers must exist somewhere in the network in order for OS/2 Warp clients to make use of the DFS.

Together with the announcement of IBM DCE for Windows NT, IBM also announced its intension to provide DCE client services for the Windows 95 platform and to deliver Directory and Security Services (DSS) as part of the IBM Software Servers for the Windows NT platform at a later time.

4.1.2 Implementation Specifics

IBM DCE for Windows NT is based on OSF DCE Release 1.0.3, except the DCE and DFS client for OS/2 Warp, as part of the IBM DCE for Windows NT package, which is based on OSF DCE Release 1.1.

An important fact to notice is that Microsoft's RPC (MRPC) is not compatible with the DCE RPC at the API level. IBM DCE for Windows NT, however, solves this incompatibility problem by providing a true DCE RPC API. It even allows you to use MRPC at the same time, for example if an application also uses OLE, which in turn uses MRPC.

MRPC is, however, compatible on the communication level with DCE RPC; thus, interoperability of DCE applications and services between Windows NT and other platforms is granted.

4.1.3 Ordering Information

IBM DCE for Windows NT can be ordered from IBM using the following order numbers:

Order Type Number	Feature Number	Part Number	Description
U.S., Canada: 5801-AAR	1721	84H4855	IBM DCE for Windows NT, V1.1, Program Package
Others: 5639-B51	5895		
U.S., Canada: 5801-AAR	1722	84H4866	Digital DCE Runtime Services for Windows NT, V1.1, Program Package
Others: 5639-B51	5896		

Table 11. IBM DCE for Windows NT Ordering Information

4.2 Microsoft Windows 3.X, Windows 95, and Windows NT (Non-IBM)

Gradient Technologies Inc. offers a product called *PC-DCE for Windows* for Windows 3.1 or 3.11, for Windows 95, and for Windows NT, respectively. It is derived directly from OSF source code and therefore fully compliant with OSF DCE. PC-DCE for Windows incorporates all DCE core functionality, for example threads, RPC, security client, directory, and time client. The implementation uses a Windows-compliant dynamic link library (DLL) that is loaded only on demand and is shared among all Windows applications. The product packaging and detailed implementation may differ among the three Windows versions. Support for OSF DCE Version 1.2.1 is available for Windows 95 and Windows NT only (PC-DCE Version 2.0).

Gradient's implementation provides its own DCE RPC because Microsoft RPC (MRPC) is not compatible with DCE RPC. Thus, DCE applications conforming to the DCE RPC standard will be portable to Gradient's DCE.

On the Windows NT platform, which supports native kernel threads, DCE threads are mapped on to the kernel threads. PC-DCE for Windows supports TCP/IP, UDP/IP, and SPX protocols.

In addition, a DCE Security Server and Cell Directory Server implementation, as well as Distributed File System (DFS), is also available from Gradient for Windows NT.

Application development for the Windows platform is supported through the Application Developer's Kit (ADK) from Gradient.

Gradients latest version, 2.0, of PC-DCE for Windows is an implementation of OSF DCE Version 1.2.1. It supports Windows 95 and the latest version of Windows NT Workstation and Windows NT Server.

Digital Equipment Corp. also offers a DCE implementation, called *Digital DCE V1.1 for Windows NT*, for Windows NT on the Intel and its proprietary Alpha hardware platforms. For Windows 95, DEC has DCE implementations available, too. Unlike other DCE implementations, DEC's DCE uses Microsoft RPC, but provides mapping functions to DCE RPC. DEC's DCE for Windows NT supports DCE core services, such as DCE Security and Cell Directory Server.

4.3 Apple Macintosh

Gradient Technologies Inc. offers *Mac-DCE* for the Macintosh, which is an implementation of the DCE core client services, based on OSF DCE 1.1. It comprises POSIX threads and DCE RPC as well as client support for CDS, DTS, and security services. In addition to this Runtime Services Kit (RTK), Gradient offers an Application Developers's Kit (ADK), necessary to develop Mac-DCE applications on the Macintosh platform.

4.4 Novell NetWare

There is no DCE implementation or interoperability solution available at this time. Some integration of NetWare and DCE/DFS on the file system level is done in OSF DCE 1.2.1, similar to the DFS/NFS gateway delivered as part of OSF DCE 1.1.

See also 9.2.1, "Integration with Other Environments" on page 139.

4.5 UNIX World

DCE is available on all major platforms in the UNIX world. We will not list them exhaustively, but the most important ones are:

- Bull S.A. offers a full DCE clients and server and DFS implementation for its AIX-based DPX/20, ESCALA, ESCALA PowerCluster, and ESTRALLA system platforms.
- For its HP 9000 family of HP-UX based systems, series 700 & 800 with HP-UX 9.0, Hewlett Packard offers *HP DCE/9000 and HP DFS/9000* with full DCE client, security, and directory server, GDS and DFS servers.
- Digital Equipment Corp. has DCE core functionality, DCE security and directory servers, available for its OSF/1 for Alpha AXP platform, including a developer's toolkit (*DCE V1.3 for Digital UNIX*). DFS server support is available through *DCE DFS for DEC OSF/1*.
- Hitachi, Ltd. offers DCE secure core services, DCE Security and Directory Servers on its UNIX-based 3050 Workstations and 3500 Servers.
- Santa Cruz Operation, Inc. (SCO) offers DCE core functionality, DCE Security and Directory Servers on its Intel-based Open Desktop 3.0 and SCO Open Server 3.0 platforms. This is a full implementation of DCE provided by SCO. DFS is not available.
- Fujitsu, Ltd. offers *UXP/DS DCE* for its DS/90 7000 series of Unix systems. Products include DCE core functionality, DCE Security and Directory Servers and an application development environment.
- Data General offers a full DCE implementation, including DCE Security and Directory Servers, for its AViiON systems.
- Gradient Technologies, Inc. offers a full DCE clients and server integration package, including an application developers kit (ADK) for UNIX System VR4 on Intel platforms. DFS is not supported.
- Transarc Corporation offers DCE implementations of all DCE clients and servers on other platforms, such as SunOS4.1 and Solaris.

DCE or some of its components is also provided by Sony, Siemens, AT&T, Pyramid, Silicon Graphics, NEC, and others on their own UNIX platforms.

4.6 Other Operating Systems and Platforms

DCE becomes more and more available on proprietary operating systems. This allows easy integration of legacy solutions in a client/server model and development of new applications that are highly vendor - and operating system-independent. The most noticeable are:

- Digital Equipment Corp. offers *DCE for OpenVMS* on VAX and Alpha AXP. This is nearly a full implementation of DCE. DTS server and DFS are not available.
- *HP DCE/3000*, provided by Hewlett Packard, is an implementation of DCE on HP/3000 platforms running the MPE/iX operating system. It includes DCE clients services, DCE Security and Directory Server. DTS server and DFS are not available.
- Hitachi, Ltd. offers DCE core services, DCE Security and Directory Servers for its OSF/1-based platform on its mainframes.

- Cray Research offers *Cray Research DCE Client Services 1.1* for a variety of its platforms. It comprises threads, RPC, security, and directory clients, DFS client and a DFS/NFS secure gateway. A DFS server (*Cray Research DCE DFS Server 1.1*) is also available.

DCE or some of its components is also available, announced, or planned for many other platforms, including:

- Bull S.A. DPS systems family
- Tandem Computers NonStop Systems
- Stratus XA/R family of fault-tolerant systems
- Siemens BS2000/OSF mainframes

Chapter 5. Security in a Distributed Environment

Customers are very sensitive to security issues. They need a strong protection mechanism for their data as well as for their transactions. Requirements and state-of-the-art mechanisms for their solution are:

- Authentication for a strong identification mechanism for user, workstations, and programs
- Authorization and permission checks with Access Control Lists (ACLs) to protect resources from unauthorized access
- Integrity to ensure network messages are not tampered with on their way to the receiver, which is achieved with encrypted checksums
- Privacy (encryption) to prevent the ability to tap the communication line and read the messages
- Auditing to track down what happened, for instance, if there were attempts to break in

Another requirement for the convenience of their users is to have a single, corporate-wide signon that is valid for all systems of an enterprise.

Many strategic applications or closed proprietary environments used to have their own way the make their operation more or less secure. They might provide absolutely perfect and secure solutions but only for their limited environments. However, customers now are looking for global, heterogeneous, and enterprise-wide solutions for the above-listed issues.

DCE offers a strong base for implementation of enterprise-wide security. It fulfills all requirements. OSF DCE 1.1 (and all its successive versions) is supplying a *Generic Security Service API* (GSS-API) that allows operating systems or other non-DCE RPC applications to use and rely on DCE security for authentication and authorization. Applications will use the GSS-API to authenticate themselves or users.

In the following sections, we discuss:

- Generic Security Service Application Programming Interface (GSS-API)
- Access Control Lists (ACL)
- Encryption

For auditing capabilities of OSF DCE 1.1, see "Auditing" on page 137. For login integration of DCE with each of the IBM operating systems, see Chapter 3, "DCE Implementations on IBM Platforms" on page 21.

For more comprehensive information about security, IBM employees can contact the IBM EMEA Security Center of Competence (CoC) in Munich through the IBM Intranet at http://w3.munich.ibm.com/CoC-Security/. This Security CoC has documents and white papers about security available that can be passed on to customers. The e-mail address is cocsecur@munivm4 for VNET users, coc-security@munich.ibm.com (IBM internal Intranet) or coc-security@ibm.de (external Internet).

Other excellent sources of information concerning security are the redbooks *Elements of Security: AIX 4.1,* GG24-4433, and *The Library for Systems Solutions Security Reference*, GG24-4106.

5.1 Generic Security Service Application Programming Interface (GSS-API)

The GSS-API is standardized by the Internet Engineering Task Force (IETF) and is described in RFC 1508. As shown in Figure 27, applications call the GSS-API, and the underlying security service is, in theory, pluggable.

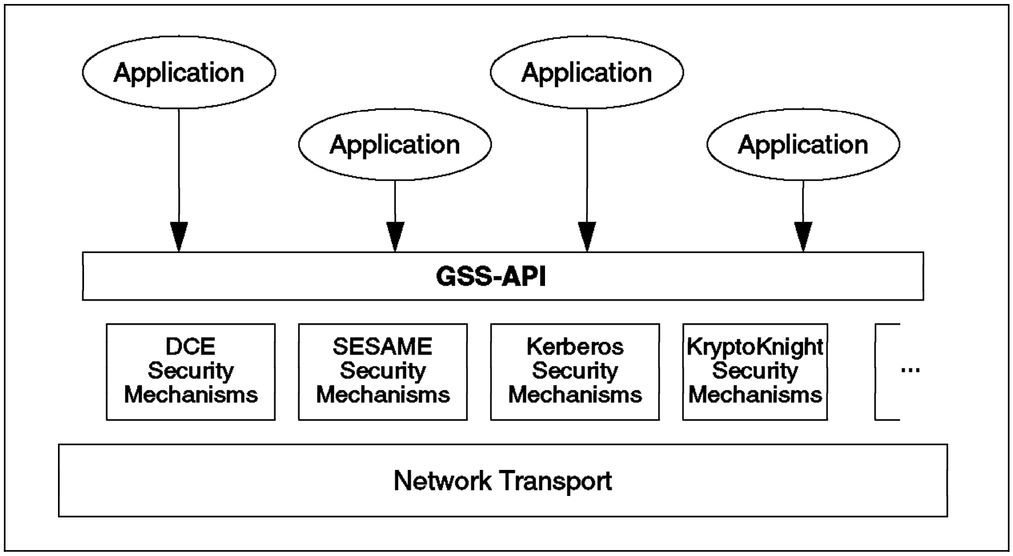


Figure 27. Generic Security Application Programming Interface

The primary goal of GSS-API in DCE is to provide non-DCE-RPC applications with access to the following DCE Security Services:

- Authentication: Allows client applications and remote resource managers to authenticate one another
- Security Context Management: Allows users and remote resource managers to build *secure connections* on top of insecure transport or association services
- Message Integrity and Protection: Allows applications to protect the integrity and privacy of data that flows in the distributed environment

A GSS-API caller (usually the application client) accepts tokens provided to it by its local GSS-API implementation and transfers the tokens to a peer (usually the application server) on a remote system. That peer then passes the received tokens to its local GSS-API implementation for processing.

Clients as well as servers first have to authenticate themselves with the Security Server (network login). Clients then use the `gss_init_sec_context()` call to create the token needed to access a server. In the DCE environment, this call results in a request to the privilege service for a service ticket to the specified server, presenting the client TGT (ticket granting ticket) obtained from the login. This ticket contains the DCE Extended Privilege Attribute Certificate (EPAC) and is returned to the client within the GSS-API security context token. The client sends the token to the server, which hands it over to the GSS-API. The server-side GSS-API verifies the client security context and keeps it for later references until it is destroyed.

In addition to exchanging tokens for initial authentication, the application (client or server) can then create and validate tokens, so-called signature tokens, to protect messages. The sender of a message requests a signature token from the GSS-API and sends the token together with the message. The receiver calls the GSS-API to verify the signature token along with the message.

A typical GSS-API caller uses its own communications protocol, calling the GSS-API in order to protect its communications with the underlying security mechanisms.

5.2 Access Control List (ACL)

An Access Control List (ACL) is a list of access control entries that protects an object. Each entry in the ACL specifies a set of permissions. Usually, most of the entries in the ACL specify a privilege attribute and a set of permissions that may be granted to the principal(s) possessing that privilege attribute. Other entries specify a set of permissions that may *mask* the permission set in a privilege attribute entry.

Privilege attributes are the user (principal) name and the name of groups of which that user is a member. The privilege attributes are passed to the server within the service ticket in the form of the EPAC.

Every ACL is managed by an ACL manager type. An ACL manager determines a principal′s authorization to perform an operation on an object by reading the object′s ACL to find the appropriate entry (or entries) that matches some privilege attribute possessed by the principal. If the type of access requested by the principal is one of the permissions listed in the matching entry, then the ACL manager type allows the principal to perform the requested operation. If the requested permission is not listed in the matching ACL entry, or is denied by a mask, permission to perform the operation is denied. Permission to perform the operation is also denied if the ACL contains no matching privilege attribute entry.

Unlike UNIX system file permissions, DCE ACLs are not limited to the protection of file system objects (that is, files, directories, and devices). ACLs may also control access to non-file system objects, such as the individual entries in a database. However, to do so, a specific ACL manager program has to be written that conforms to a predefined ACL manager interface.

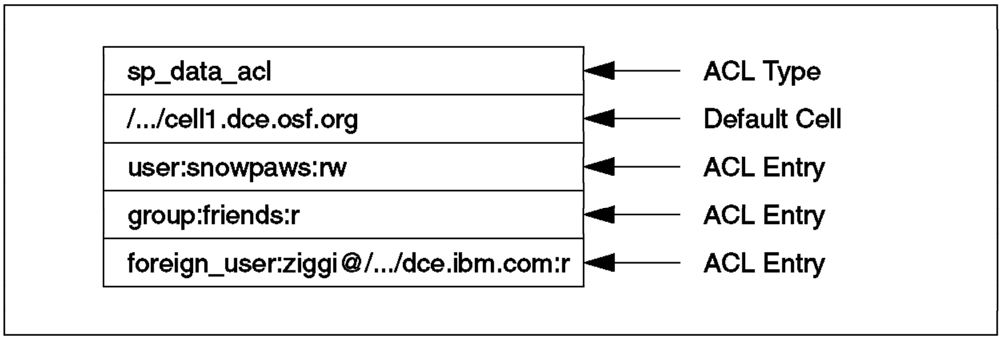


Figure 28. DCE ACL Example

An ACL consists of the following:

- An ACL manager-type identifier that identifies the manager type of the ACL
- A default cell identifier which identifies the cell where a principal or group identified as local is assumed to be a member

- At least one ACL entry

DCE authorization defines two types of ACL entries:

- Those that associate a specified privilege attribute with a permission set — these are privilege attribute entries
- Those that specify a permission set that masks a permission set specified in a privilege attribute entry — these are mask entries

In Figure 28 on page 89, if user *snowpaws* tries to access the object, he/she may read it or write to it. Members of group *friends*, and user *ziggy* of a foreign cell, may only read the object. Any other users have no access.

5.3 Privacy

When a client establishes authenticated RPC, it can specify the level of protection to be applied to its communication with the server. The protection level determines the degree to which client/server messages are actually encrypted. As a rule, the more restrictive the protection level, the greater the impact on performance.

The following protection levels are available:

None No communication protection.

Connection Performs an encrypted handshake the first time the client communicates with the server.

Call Attaches an encrypted verifier only at the beginning of each remote procedure call over connectionless communication. This level does not apply for TCP connections.

Packet Attaches a verifier to each message sent over the network to make sure all messages are from the expected client.

Packet Integrity Ensures and verifies that no messages have been modified by computing and encrypting a checksum over each message.

Packet Privacy Encrypts RPC arguments and data in each call.

Encryption is done with the session key, which is only known by the client and the server for which the service ticket was issued.

On most platforms, encryption is done with the data encryption standard (DES) algorithm, which cannot be exported outside the US in a user accessible form. This means that DES can be used for protection levels up to and including *Packet Integrity*, but not for *Data Privacy*.

On the AIX platform, there is a User Data Masking Facility, which is still referred to as Common Data Masking Facility or CDMF. CDMF allows you to encrypt user data in RPCs using DES with a 40-bit key instead of the standard 52-bit key. Since this makes the encryption weaker, it has less export restrictions from the US. It is a good solution for non-US customers who want increased privacy, but cannot have an export license for full DES.

Chapter 6. IBM′s Software Products Will Exploit DCE

DCE is IBM′s strategic distribution service and one of the strategic communication services. This certainly means that new products will be implemented using the suggested layers. There are already a number of products on the market that were designed for and built on top of DCE, such as DFS or Encina and CICS for AIX, IBM′s strategic products for distributed Online Transaction Processing (OLTP). The Open Group (former OSF) compiles and publishes a list of DCE applications from any vendors. See Appendix A, “Other Sources of Information” on page 147 for information on how to get this catalog.

On the other hand, IBM′s commitment to DCE in the Open Blueprint also means that existing products will evolve to incorporate these layers and services. This will make them more interoperable, or hardware independent, and provide them a stable management infrastructure with global security, time, directory, and transaction services. Other products, such as LANDP, receive mapping capabilities to allow interoperation with DCE servers and clients.

In order to stress again the importance of DCE in IBM′s product strategy, we examine some known and existing products and explain how they will use DCE functions.

The fact that these products are receiving a DCE layer does, in most cases, not affect the end user of the product or application data; it just changes the behavior of the product behind the scene and enhances the capabilities of the product. The ones who notice the DCE layer are the administrators. They will have to deal with DCE and understand it well to be able to maintain it. However, for them, the advantage is that administration of the other products is simplified and unified.

The first step of DCE integration for these products is to use the directory service to make data or services globally available and easier to administer. A second step would be the integration with the DCE Security Service to allow for corporate-wide authentication (single signon) and security policies. A next step would be to either directly use secure DCE RPC or to embed existing, weaker protocols in DCE RPC.

The products described in the following sections have implemented one or the other of these steps, or will do so in future releases.

Note: The remarks about future DCE capabilities of the products mentioned hereafter are neither product announcements nor formal statements of direction. They are collected from presentations made by IBM speakers at trade shows or press conferences, which provides them some degree of officiality. However, there is no guarantee if and when they might be implemented.

6.1 IBM OS/2 Warp Server and OS/2 LAN Server

The IBM OS/2 LAN Server and OS/2 Warp Server family of products provides a powerful network operating system for data and resource sharing among client and server machines in a network. LAN Server servers run on OS/2, OS/400, MVS, or AIX. These LAN Server implementations provide service to LAN Server requesters or clients. Some of them even support clients of competitive products, such as Microsoft′s LAN Manager and Novell′s NetWare. LAN Server clients run on DOS, Windows, and OS/2.

LAN Server 4.0 has received high praises and market acceptance for its simple installation and user-friendly graphical administration interface. With the announcement of *IBM Directory and Security Server for OS/2 Warp, Version 4*, IBM OS/2 Warp Server and IBM OS/2 LAN Server have been enhanced by using DCE services underneath.

OS/2 Warp Server and OS/2 LAN Server use DCE Directory and Security Services in place of the analogous LAN Server functions. To accomplish this with a minimum of impact on the administrator, the popular new graphical interface introduced with LAN Server 4.0 has been extended logically to handle the DCE-specific components.

The new DCE directory and security facilities are implemented by means of an OS/2 LAN Server or OS/2 Warp Server feature that is installed with DSS for OS/2 Warp on at least one domain controller in the network. This feature contains a DCE client. A DCE Directory and Security Server are then installed on any machines in the network. OS/2 Warp Server and OS/2 LAN Server can use any OSF DCE servers as long as they are based on OSF DCE 1.1. For the sake of easy handling, the DCE servers for OS/2 are packaged with the Warp Server and LAN Server feature and form the IBM Directory and Security Server for OS/2 Warp, Version 4, product.

There is a migration tool that moves the user and directory data from the LAN Server domain controller database (DCDB) and NET.ACC file to the DCE Cell Directory and Security Server database. Multiple domains may be migrated to the same DCE cell, enabling easy cross-domain resource sharing. It is important to note, with this approach, that the multiple domains effectively become a single administrative entity—a DCE cell. Therefore, user names must be unique within the larger domain. Collisions need to be handled manually during the migration process. With the DCE Cell Directory Service, the user has only one identity and password that is known to all the domains that are migrated to that cell. All of the resources in the cell, or in peer cells, are accessed under a single login. There is also a synchronization mechanism that ensures that the DCE directory and security databases stay in synchronization with the legacy LAN Server databases. This is done so that 3.x legacy clients can run unchanged against the Warp or LAN servers.

Clients use the DCE directory to find objects in the network, such as printers, files, and so forth. Legacy clients, which cannot support DCE global names, must define an alias that points to the DCE global name of a resource in order to find it on the network. DCE is used for authorization on Warp or LAN servers. Clients use DCE Kerberos for authentication. Legacy clients use the existing LAN Server authentication mechanism.

There are two types of clients. The integrated client contains both a LAN requestor and a DCE client. The unintegrated client contains only a LAN requestor. At least one integrated client must be installed to manage the LAN Server and DCE pieces of the network. Additional clients may be installed. Customers who are writing applications to the DCE RPC model can use the integrated clients to achieve a single logon across DCE applications and the file and print sharing LAN Server functions.

The following are the key benefits to the new OS/2 Warp Server and OS/2 LAN Server products using DCE services from IBM Directory and Security Server for OS/2 Warp, Version 4:

- DSS for OS/2 Warp contains an open, scalable Global Directory Service. This new directory model has been integrated with the existing LAN Server product to provide support for current environments and interoperability across multiple vendors' DCE service providers.
- LAN Server′s file and print serving performance has been rated by independent test laboratories as superior to that of any other LAN file/print server. Now in its fourth major release, OS/2 Warp and OS/2 LAN Server are used in more large enterprises than any other LAN file/print server.
- DSS for OS/2 Warp provides open, distributed application support for those customers using the OSF DCE RPC and the integrated clients. Because of this support, distributed applications can transparently take advantage of Kerberos third-party authentication and DCE′s global directory support.
- OS/2 Warp Server and OS/2 LAN Server, Version 4, contain a state-of-the-art GUI administration interface that has been extended by the IBM Directory and Security Server for OS/2 Warp, Version 4, to not only manage Warp and LAN Server domain resources but also any DCE servers anywhere in the enterprise. This GUI is an extension of the award-winning LAN Server 4.0 administrator interface.
- OS/2 Warp Server and OS/2 LAN Server, Version 4, allow existing, unchanged LAN Server clients to fit into the new global directory via the synchronization mechanism. This means that clients can be upgraded over a period of time, when and if the customer wishes.
- The OS/2 Warp Server and OS/2 LAN Server, together with the IBM Directory and Security Server for OS/2 Warp, Version 4, are evidence of delivery on IBM′s Open Blueprint strategy. One goal of the Open Blueprint is to provide customers access to all authorized resources with a single login.

For additional information on IBM Directory and Security Server for OS/2 Warp, Version 4, see also 3.5, “OS/2 Warp Implementation of DCE” on page 68.

6.2 IBM AIX Connections

IBM AIX Connections has been extended in its latest update to facilitate DCE interoperability. AIX Connections provides LAN requestors from an OS/2 LAN or OS/2 Warp Server, Windows 95, or Windows NT environment seamless access to AIX resources, such as AIX file access without requiring the user to log in to AIX.

The new DCE interoperability enhancement automatically logs a user in to DCE, providing him access to DCE resources, namely access to DFS files. The user does not need to log in manually to DCE, nor does his machine need to be configured as a DCE client. A user will only be required to log on to his domain controller, providing his user ID and password. AIX Connections, running on an AIX system configured as a DCE and DFS client, then logs this user on to DCE and bridges the DFS filespace to the PC, appearing as a remote drive. Full DFS access security applies since the user is well known to DCE and DFS just like any other DCE principal with its own user ID.

6.3 IBM DATABASE 2

One of the key features of DB2 on any platform is its strong client/server support. It allows client applications to access remote database servers, using either private protocols or IBM′s Distributed Relational Database Architecture (DRDA) database protocols, via any major communications protocol stack. DB2 client applications can access DB2 servers on AIX or OS/2 via the DB2 LAN protocol (DB2RA), whereas access to mainframe DB2 (MVS, VM, or OS/400) requires the DRDA protocol, which is not directly supported by these clients. So, they need access to a DB2 gateway speaking both protocols and translating back and forth. The Distributed Database Connection Service (DDCS) is required on this gateway. DDCS is available for AIX and OS/2. DB2 LAN or DRDA protocols allow you to route SQL commands to remote databases, much like, for instance, SQL*Net used by Oracle.

When clients need to access remote data, they need to locate the database that contains the required data. Databases, servers, and gateways need to be cataloged, and every client holds a copy of this catalog. In a large environment with thousands of workstations, this requirement imposes a huge work load on the database administrator.

This is where DCE comes into play. These resources can be cataloged in a single place: the CDS database. The cataloging in CDS eliminates the necessity of maintaining catalogs on every remote client. The CDS database provides a global directory for:

- Databases
- Database server instances
- Routing information (DDCS gateways)

The CDS contains objects of these three types. The administrator creates the information only once in a central location, the CDS server of a DCE cell. Through the use of DCE′s Global Directory Services, databases anywhere in the world can be located and accessed. Another advantage of this is that resources can be moved with few administrative changes.

A database instance AIXDB2 in a DCE cell /.../AUSTIN_CELL may have the following CDS entry:

```
Object name:              /.../AUSTIN_CELL/subsys/database/AIXDB2
DB_Object_Type:            L
DB_Product_Name:           DB2_for_AIX
DB_Product_Release:        V2R1M00
DB_Native_Database_Name:   AIXDB2
DB_Database_Protocol:      DB2RA
DB_Authentication:         CLIENT
DB_Communication_Protocol: TCPIP;mozart;1234
```

If a client uses the fully qualified DCE name (the object name in the above example), the access routines extract the database server instance name, which is also called a Database Management System (DBMS) instance object. From this (CDS) object, the communications protocol and all necessary parameters (for instance the host name) can be extracted to make the connection to the database. The connection is established in one of the traditional ways, such as TCP/IP sockets, NetBIOS, and so on, but currently not with DCE RPC. So, the CDS is only used to find the necessary information to route the connect request over the right link, with the right communications protocol, to the right server. If the DB client and server do not use the same DB protocol (DB2RA or DRDA), the client can extract the location of a DDCS server from a routing information object in CDS.

The use of CDS is not mandatory, and in the current release, only AIX clients will support it. DB2 clients can still hold the local catalog. If the requested database location is found in a client′s local catalog, this information is used without trying to contact CDS.

The latest version of DB2 for MVS/ESA has been enhanced by adding the capability for DRDA client/server authentication using DCE security functions. This simplifies the task of userid and password management database users and strengthens security by eliminating the need to flow MVS userids and passwords as visible text.

At trade shows or press conferences, IBM speakers have also presented that in future releases:

- All DB2 platforms will get the CDS support
- Extensions are made to the DRDA architecture to support DCE security tokens
- DCE Security Service will be used to authenticate DB2 clients to DB2 servers

6.4 IBM MQSeries

IBM MQSeries is a message queuing interface that enables applications running on different platforms to asynchronously exchange messages. It is available from IBM for many IBM and non-IBM platforms, such as AT&T GIS, Sun Solaris, Tandem, HP-UX, SCO UNIX, UnixWare, VMS, and SunOS. The current AIX version allows the use of the DCE Directory Server to locate queues and queue servers. MQSeries is described in more detail in the application development chapter under 7.5.3.1, “IBM MQSeries” on page 123.

At trade shows or press conferences, IBM has also presented that MQSeries products will take advantage of DCE Security Service in a future release.

6.5 IBM Transaction Servers

The IBM Transaction Servers, as part of the IBM Software Servers series of modular application server products, are available for different platforms, such as AIX, Windows NT, and OS/2 Warp.

6.5.1 IBM Transaction Server for AIX

The *IBM Transaction Server for AIX* includes two technology-leading transaction monitors: IBM’s *CICS for AIX* and Transarc’s *Encina for AIX*. A transaction monitor serves as an application platform, which provides highly reliable functionality for distributed, transaction-oriented applications. Such applications can rely on these functions and do not need to incorporate this level of complexity in their own code. The IBM Transaction Servers are ideally suited to build, execute, and manage three-tier applications.

A transaction is a unit of work consisting of one or several single tasks on one or more computers that needs to be executed all together or not at all. A classical example of a transaction is a fund transfer from one bank account to another. It consists of a debit action on one account and a credit action on the other. For obvious reasons, it would cause a serious problem if only one action took place because a computer was not available at that point in time. In order to guarantee data consistency at any point in time, applications make use of transaction monitors to carry out such kinds of distributed, transaction-oriented processing. Many transaction-oriented applications involve databases on different systems, both for query and update operations.

Transaction monitors themselves need to know the location of their remote counterparts as well as security features that protect transactions from being tampered with. Both, IBM CICS for AIX and Encina for AIX use DCE for security and directory lookup.

IBM CICS for AIX is the AIX implementation of IBM's traditional transaction monitor, being developed and constantly enhanced on mainframes for over 20 years. It ensures interoperability with all other platforms supported by CICS. CICS clients are available for OS/2, DOS, Windows 95, Windows NT, Macintosh, and Sun UNIX systems. The AIX implementation of CICS can make use of the following DCE services:

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

Although CICS can use its own directory and security mechanisms, it is recommended to use DCE's services instead. Most often, CICS for AIX is used in combination with and makes use of some modules of Encina for AIX.

Encina for AIX may be viewed as a logical extension to DCE that provides a higher level of application programming interface, allowing efficient an cost effective application development even in environments where transaction integrity is not a primary requirement. However, it also offers transaction-oriented services on top of DCE and is therefore well suited for applications where data integrity is a major issue.

Customers may decide on a per-project-basis on whether to use CICS or Encina, as both offer similar functionality. CICS for AIX is the transaction monitor of choice if a customer already has remarkable investments in CICS technology and skills on a mainframe and wants to continue using this technology and spread it to distributed AIX systems. Encina on the other hand could be favorable if a customer has not already been using other transaction monitors, or if it is used as a productivity-increasing tool for development of client/server applications using DCE.

Additional information on using IBM CICS for AIX and Encina for AIX as a high-level application programming interface can be found in 7.5.3.2, “Encina” on page 123, and 7.5.3.3, “IBM CICS” on page 124.

6.5.2 IBM Transaction Server for Windows NT

Similar to the IBM Transaction Server for AIX, the IBM Transaction Server for Windows NT, as a new member of the IBM Software Server Series, contains two major parts: *IBM CICS for Windows NT* and *Encina for Windows NT*. Clients included in the product are:

- CICS Clients for AIX, Windows 95, Windows NT, and OS/2
- Encina Clients for AIX, and Windows NT
- DCE Client for Windows NT (DCE Runtime Services for Windows NT)

Both CICS and Encina make use of DCE for Remote Procedure Calls (DCE RPC) and Threads Services. CICS for Windows NT uses the services of Encina and provides a range of transactional services for developing and deploying distributed transaction processing systems.

Encina for Windows NT may be viewed as a logical extension that adds value to the DCE environment, offering productivity improvement and supplementing DCE services. Its client/server functions 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.

The IBM Transaction Server for Windows provides a powerful application development environment for transaction-oriented applications, similar to those of the IBM Transaction Server for AIX. See 7.5.3.2, “Encina” on page 123, and 7.5.3.3, “IBM CICS” on page 124 for more information.

6.6 IBM Single Sign-On

Although not formally announced at the time of writing this publication, IBM has intended through press releases to introduce a product called *IBM Single Sign-On* (SSO), sometimes also referred to as *IBM Secure Single Sign-On*. This key piece of technology is intended to solve one of the most often mentioned end-user requirements: It allows a user to log on to as many systems and/or applications as required by entering his/her user ID and password only once on a single dialog.

SSO uses open standards: It uses DCE base technology for secure communication, and DCE Security Services for user and password registry services. SSO will be supporting the following, most popular current clients, allowing users on these systems to participate in a large complex of SSO-integrated systems:

- AIX
- OS/2 Warp
- Windows 95
- Windows NT

SSO is able to log users on to a variety of systems and applications:

- Local Operating Systems (AIX, NT)
- IBM LAN Server 3.X and 4.X, and Warp Server
- Novell NetWare 3.X and 4.X
- Microsoft NT Server
- Mainframe applications protected by RACF

- DCE applications
- Databases (DB2)
- Any other application with enablement

Upon user log-on, the SSO client software will communicate with a corresponding SSO service on any target system (see above) that interfaces to the specific products and logs the user in. Since secure DCE communication is used, no password will ever be transmitted in the clear to any other system. SSO uses either locally stored, encrypted passwords on the target systems, or it uses generated one-time passticket technology, depending on customization. In either case, password changes are carried out by SSO on user request.

SSO provides a graphical user interface to the user for login and for a selection of applications this user is authorized to access.

DCE runtime software is required on any client, as well as on the corresponding application client, such as LAN Requestor or Novell NetWare client that the user is about to use. Utilities available with SSO allow easy migration from current NetSP SLC registered users to the DCE Security Services database used by Single Sign-On.

SSO will require an AIX DCE Security Server somewhere in the cell. Support for other security servers, such as DSS for OS/2 Warp or DCE on OS/390, is planned in a follow-on release. Future releases may also support smartcard technology and light-weight DCE clients. Further target system support, such as for ACF2, TopSecret, and OS/400, is another improvement perspective in future releases of SSO.

6.7 IBM DFS Web

IBM DFS Web brings DCE DFS security and scalability to the Web. As security—basically who is allowed to have access to which data—becomes one of the most important issues in today's Intranet and Internet applications, DFS is just an ideal file system to store the data for Web applications.

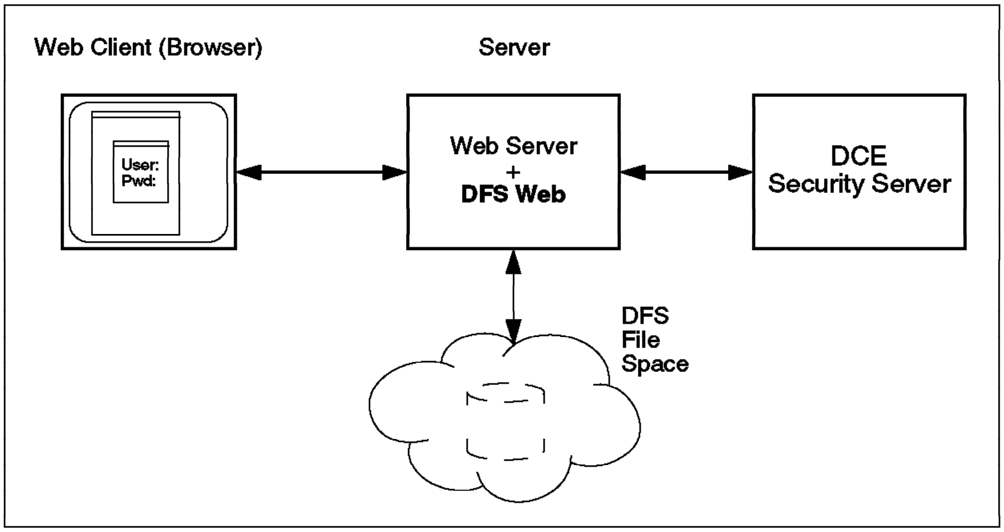


Figure 29. DFS Web Exploits DFS Security and Scalability

By using DFS to store the Web data (HTML files, pictures, data for CGI programs, and so forth), as illustrated in Figure 29, instead of using native UNIX file systems, three substantial advantages are added to the solution:

1. DFS security is superior to conventional UNIX file system security. Users can be granted selective access, either individually or by grouping them together into groups with equal access permissions, allowing for easier administration.
2. Data can be stored anywhere in the DFS file space, indicated as a "cloud" in Figure 29 on page 98, and does not need to be copied to a dedicated storage area on the Web server. This allows for a maximum of scalability and availability since DFS data can be replicated.
3. Any conventional Web browser can be used on client machines since security is handled only on the server side. No plug-ins or any additional pieces of code are necessary for these browsers.

DFS Web is implemented as a NSAPI (Netscape Server API) plug-in to an existing Web server. An additional administrator GUI allows for easy user and group administration as well as ACL administration for the data access permissions. DFS Web also includes an Account Manager for auditing, logging, and even for alerting administrators in certain circumstances.

When a user on a Web browser first tries to access protected data in DFS, he/she will be presented a dialog for user ID and password. DFS Web will convert these into DCE credentials giving the user the specific access permissions on DFS data.

DFS Web is planned to be available from IBM in the first quarter of 1997.

Note:

At the time of writing this publication, DFS Web has only been pre-announced by IBM through press releases, which are not formal product announcements. The product's name and implementation are still subject to change until official announcement.

Simple DCE DFS Security for the Web: DFS access security can even be used for protecting the Web without any additional software just by using standard Web servers and clients (browsers), together with DCE and DFS. With a client machine serving as a DFS client, a user will log on to DCE, getting his/her DCE credentials and access rights to the DFS file space. Provided the Web pages are built such that they access local files instead of files stored on the server, and provided these local files are stored in the DFS file space, the same level of security can be achieved as when using the DFS Web product just described. Remember that "local" means software on a client system can access a file as if it is stored on the local disk. DFS makes files look like if they are stored on the local disk although they are stored somewhere on a DFS server.

The drawbacks of this solution, compared to DFS Web, would be that every client system needs to be a DCE and DFS client. Also, there is no such auditing as with DFS Web, and finally, there is no central, GUI-based administration.

6.8 IBM LAN Distributed Platform (LANDP)

LANDP is a multiplatform (DOS, OS/2, OS/400, and AIX) client/server solution that was originally called Financial Branch System Services (FBSS). As the finance industry evolved from 3600 and 4700 systems to intelligent workstations (PCs, PS/2s, and RISC/6000s), the FBSS family of products (FBSS/DOS, FBSS/2, and FBSS/400) was created to support client/server processing in the intelligent workstation environment. As the focus of FBSS expanded beyond the financial arena, the products′ names were changed to LANDP (LANDP/DOS and Windows, LANDP/2, and LANDP/400) and expanded to include the AIX operating system (LANDP/6000). The LANDP products make it possible to share resources and services between all the components running in a heterogeneous LAN environment that contains the different operating systems.

Besides offering a LAN-based client/server platform, LANDP is an application enabler, providing a common and consistent application programming interface (API) that is used by all LANDP resources. Several services are available that allow the implementation of financial transaction systems through the LANDP API. Some of these services are:

- SNA Server

  This allows client applications using the LANDP common API to request SNA services and conduct LU 0, LU 1, and LU 2-type SNA sessions to host systems.

- Program-to-Program Communications (PPC) Server

  This enables client applications using the LANDP common API to communicate with a partner client application using an SNA LU 6.2 session.

- Query Server

  This translates database requests from the LANDP common API format into SQL requests and sends them to the database. By using the LANDP common API, the complexities of indexed file access and dynamic SQL queries are hidden from the client application.

- Electronic Journal Server

  This allows applications to create, manage, and process data in electronic journals that are used to maintain a record of all processed transactions for transaction recovery or auditing.

- Store-for-Forward Server

  This enables client applications to store and manage transaction records that are destined for the host when the host connection is not available. This allows for offline transaction or transaction queueing. The records can then be sent to the forwarding server.

- Forwarding Server

  This service enables for store-for-forwarding or queued records to be sent to the host when the connection is reestablished.

- Shared File Server

  This service provides a shared file database that allows client applications to store, retrieve, modify, and delete records using the following access methods:

  - Direct access

  - Sequential access
  - Indexed access
  - Indexed sequential access

- Various Financial I/O Servers

  These servers provide the ability to share financial I/O devices (printers, magnetic stripe readers, PIN pads, automatic teller machines, and personal banking machines) between multiple clients.

The LANDP platforms use a common application programming interface that relies on the remote procedure call paradigm for communication between clients and servers. Through the use of this common API, LANDP allows you to create client applications and new server applications.

The LANDP-DCE service of the LANDP/6000 product allows clients and servers written with DCE RPC to interoperate with the LANDP clients and servers. This LANDP-DCE server is referred to as a *mapper*.

This mapper resides on a RISC System/6000 that has both LANDP/6000 and IBM AIX DCE installed. The mapper converts requests and replies from DCE RPC format to LANDP format.

When loaded, the LANDP-DCE mapper registers itself to the directory service namespace as a server with two entries and a directory:

```
/.:/subsys/LANDP
/.:/subsys/LANDP/default
/.:/subsys/LANDP/<LANDP_workgroup_name>
```

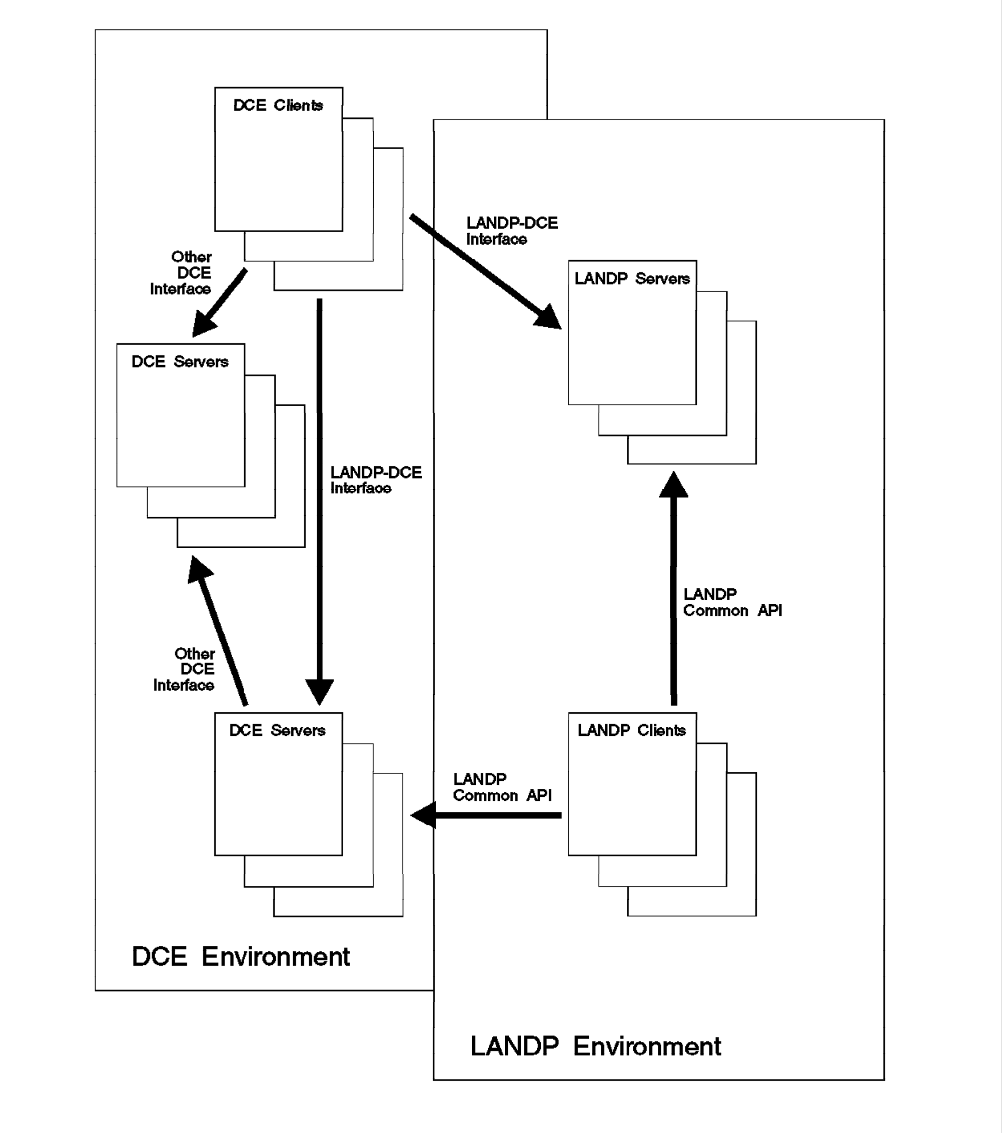


Figure 30. LANDP-DCE Interoperability

DCE clients or servers need to be written to the LANDPDCE interface definition, the landpdce.idl file, which is predefined by LANDP. Its function calls correspond to LANDP calls. Figure 30 shows how DCE and LANDP interoperate by mapping calls from one protocol to the other.

The LANDP-DCE mapper (not shown) sits between the two environments. Clients from both sides use their native protocol to talk to servers on the other side. However, instead of talking to a server, they communicate with the mapper that translates the calls and responses back and forth. The mapper itself also uses the LANDPDCE interface definition. To DCE clients, the mapper presents itself as a DCE server supporting the LANDPDCE interface and acts as a LANDP client toward the native LANDP environment. To LANDP clients, it will likewise present itself as a LANDP server that uses the LANDPDCE interface to behave like a DCE client toward DCE servers, which need, of course, to be written to support the LANDPDCE interface.

6.9 IBM Printing Systems Manager for AIX

The IBM Printing Systems Manager (PSM) for AIX is based on the Athena Palladium Project. It is a robust, standards-based print administration and management product that solves a fundamental problem in today′s printing environment. Users want total flexibility in a distributed environment. They want to be able to:

- Print PostScript, PCL, PPDS, and other formats
- Use printers from HP, IBM, Lexmark, and others
- Print data stored on Windows, Mac, MVS, OS/2, HP, Sun, DEC, UNIX, and others

The need for centrally managed printing in the heterogeneous print environment has been recognized in the print industry for some time. In the 1980s, MIT′s Project Athena created Palladium to show that the technology exists to fill this need.

The Athena system is based on UNIX 4.3 BSD. The Berkeley print spooler was written for a central processor environment. Palladium was designed to overcome the shortcomings of the Berkeley spooler and thus satisfy the requirements, such as:

- Placing printers, users, and spool queues all on different machines in a networked environment
- Support for different types of printers and formats
- Easier support for centralized management
- Authentication of users, not just from an /etc/passwd file
- Access control, which was not necessary before
- Accounting and logging

The design of the original Palladium print system has been realized through the application of the following distributed-system concepts:

- Client/server model — Proved very useful for the relationship between local user applications and shared resources in the distributed environment. This approach allowed many clients to obtain central services and facilitated central management.
- Name service-based server advertisement — Allows the clients to easily obtain the address and information necessary to get print service. Facilitates administration because printers need to be defined in the location service only and could be easily moved.
- RPC-based server access protocols — Palladium clients and servers use the RPC paradigm for communication.
- Print server database — Stores all queued print jobs and print-system configuration. Is accessed by the job scheduler and by administrative functions.
- Print job management — A job scheduler on a server locates the right printer to satisfiy the job requirements, schedules the job on the printer, and tracks the progress of the job. The server collects events reported to it by the supervisors in the job info database, which is part of the server database,

and reports status and significant events to the client. Accounting and status information are logged.

- Authentication and authorization — For each request made by a client, the server checks a well-known access list to determine if the user is authorized to make the request. Access lists are maintained for each operation. Secure authentication of the client′s identity is performed by Kerberos.

The above description of the original Palladium implementation matches exactly what DCE provides. Palladium was made before DCE was born. The *IBM Printing Systems Manager for AIX* integrates this functional requirements with DCE.

PSM uses DCE Security, Time, RPC and Directory Services. PSM fully relies on and 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 supports the IBM Open Blueprint for distributed computing.

PSM allows customers to:

- Manage their network printers remotely with a GUI from any AIX client

  Printers are defined at a central place and can be managed with a sophisticated GUI from any AIX client workstation. This makes it easy to redefine the printer network at any time.

- Monitor and configure the print network from two types of interfaces:

  - An administrative GUI
  - A POSIX command line interface

- Implement authentication and control of user-access to printers and functions

- Use their existing printers, applications, and print commands

  The new product will support printers that are currently supported in the AIX environment and by IBM′s PSF/6000. The users (and applications) can still submit, query, and cancel print jobs with the very same commands, applications, and drag/drop mechanisms they use today.

- Manage production printing and on-demand printing

  Integration with PSF/6000 enables users to take advantage of the capabilities of Advanced Function Printing (AFP).

- Use open interfaces

  PSM provides the common *lpr* interface that allows users on, for example, OS/390, VM, Sun OS, Solaris, DEC Ultrix, and HP-UX systems to make use of the advanced distributed printing capabilities it provides.

- See events when they happen

  Asynchronous event notification will inform specific people that they need to take action.

- Migrate and expand at their own pace

  The system can be easily expanded to accommodate more printers, servers, and clients.

6.10 DCE Distributed File System Exploitation

DCE DFS is a distributed file system with some clear advantages over other network file systems, like:

- Superior performance—As it has been evaluated in many benchmarks, DFS outperforms other network file systems for most types of applications.
- Reliability—Due to its log-based recovery/restart capabilities, similar to those found in database products, DFS proves to be very stable and robust.
- Replication—DFS filesets can be replicated to multiple server machines and thus provide a built-in mechanism for high-availability.
- Caching—DFS's excellent caching capabilities allow faster file access and less network and server load by maintaining full file integrity due to its unique token-management.
- Security—DFS provides access control, based on DCE's security mechanisms. File access can be managed on a per-user or per-group basis and provides more permission attributes than the standard-UNIX filesystem.

For these reasons, DFS is often being used even in installations where the features of DCE is not a primary requirement.

6.10.1 DFS for Application Development Environments

Software development environments are typically centered around a common base of files that need to be accessed by many client systems. Software developers enjoy the power and freedom with their own workstations, but they need fast access to common source files and libraries. Compile and link operations are file-intensive tasks that require powerful file input/output capabilities for both read and write.

DFS is a perfect distributed file system for this kind of file-sharing work. Its excellent file write performance allows for shorter compile and link times.

6.10.2 DFS for CATIA

CATIA, a leading Computer Aided Design and Computer Aided Manufacturing (CAD/CAM) application from Dassault Systemes, has been successfully used with DFS as the common file system in several large installations. CATIA usually runs on high-end, graphic-oriented workstations and needs fast access to a large amount of common shared data, like parts catalogs and common development libraries.

Due to its performance, robustness and scalability, DFS helps to keep and increase the productivity of the engineers working on these CATIA workstations. DFS has proven its ability to outperform other network file systems in several CATIA installations.

More information about CATIA solutions can be found on IBM's CATIA Web pages at http://www.catia.ibm.com/html/catmain.html.

6.11 Adstar Distributed Storage Manager (ADSM)

ADSM is IBM's distributed multiplatform data backup and recovery product. It supports almost any important operating system, including mainframe systems and a variety of subsystems. Automated backup policies and schemes can be implemented to save all data on distributed systems reliably on many supported mass-storage devices.

Users on client systems can retrieve data in the form of single or multiple files by request through a graphical user interface.

ADSM is capable of handling the various native file-system types on its range of different client systems, such as OS/2 or AIX.

With an enhancement, ADSM will be capable of backing up and restoring DCE/DFS data by the addition of a DFS Backup Client on AIX workstations. Handling DFS data is different from most other data because of its ACLs (Access Control Lists). ADSM preserves the ACLs with the files and directories and is aware of DFS mount points.

The mentioned enhancement, AFS/DFS Backup Client, is planned to be available by year-end 1996.

Chapter 7. Client/Server Application Development

There exist many different client/server models for application development. This chapter focuses on client/server application development with DCE functionality. This includes development of new applications and upgrades of legacy applications to DCE client/server-based applications.

This chapter also describes a number of different tools on the market that can assist the programmer in the development of DCE client/server-based applications. Some of these tools are developed by IBMs and some are from other vendors.

7.1 Why Use Client/Server Technology?

Client/server applications have become popular over the last five years. The background for client/server applications and their characteristics are described in the following section.

7.1.1 Introduction

The computer system evolution over the last three decades can be divided into three main areas.

- Central computer systems running batch jobs. This was the primary way to use computer systems in the ’70s. There was no direct user interaction with the computer systems. All computer jobs were initiated and controlled by an operator.
- Central computer systems accessed via terminals. This was the ’80s alternative to the batch-oriented way of working with computer systems. The user could then, via terminals, interact directly with the computer systems. This way of direct interaction is also called online computer systems. It became very popular and is still a very common way of connecting users to computer systems. In the ’80s, minicomputer systems also became popular. Basically, they offer same functionality as the central computer systems, but with less performance. Minicomputers are often called department servers.
- Client/server-based systems. In the late ’80s, when the PC became popular, many terminals were replaced with PCs. At the same time, Local Area Networks (LAN) were implemented. The PCs were primarily used for PC-based text processing and spreadsheet applications, but the terminals were replaced with PC terminal-emulation programs. The LANs interconnected the PCs to the central computer systems. The users could then use their PCs in a combination of running local PC applications and terminal-emulation applications. This has lead to a lot of confusion about missing integration between PC applications, central computers, and minicomputer systems. The solution was to develop client/server applications.

7.1.2 What Is Client/Server?

Client/server in computer terms means that an application program has been divided up into a client part and a server part. The client is normally the part that is running on a PC or any other desktop, and the server is normally the part that is running on the central computer or on a departmental system.

It is evident that PCs, central computers, and minicomputer systems all have different characteristics. For example, PCs have a brilliant user interface with a graphical screen and mouse input—but the computing power in a PC is not very good, compared to central computer systems. On the other hand, the user interface in central computer systems is not user friendly.

The idea is to use the best from the PC, central computer, and minicomputer and combine it into a one logical system. The goal is that the users should not be able to see that the programs they are running are client/server-based programs.

7.1.3 Communication Between Client and Server

A client/server application normally consists of a client part running on one computer system, for example a PC, and the server part running on another computer system.

To support this, a communication system must be in place on both the client and the server side that allows the client system to talk to the server system. Also a communication protocol must be in place so the client and the server have a structured way of exchanging information.

Over the years, many different communication protocols have been developed, each with its own special characteristics. These different protocols have been developed to solve specific requirements for communication between computer systems.

Popular communication protocols today are TCP/IP, NetBeui/NetBIOS, IPX, SNA, and AppleTalk, just to mention a few.

It is very important that the client and the server side have elected the same communication protocol; otherwise, they will not be able to communicate with each other.

DCE normally uses TCP/IP as the communication protocol, although some vendors support other communications protocols as well.

7.1.4 The Programmer’s Interface to the Communication Protocols

A programmer who writes a client/server-based application must bind the application to the communication system. To support this, the communication system offers an application programming interface, also called API.

Unfortunately the APIs are different between the communication protocols, and the programming methods are also different. This means that it can be a very complex job to write a client/server application when using the native APIs offered by the communication system. Also it can be a very complicated job to reprogram a client/server application to allow it to use another communication protocol.

The API offered by the communication system does not provide high-level functionality, like security and directory services. This must be implemented by the programmer on his/her own. This can be very complicated.

DCE has its own API for development of client/server-based applications. This API includes the use of Security, Directory, and other services as well as a generic way of integration to the communication protocols. The DCE API is the same among all platforms that support DCE.

DCE makes it easy to program client/server-based applications. In DCE terms, a client/server application is called a distributed application.

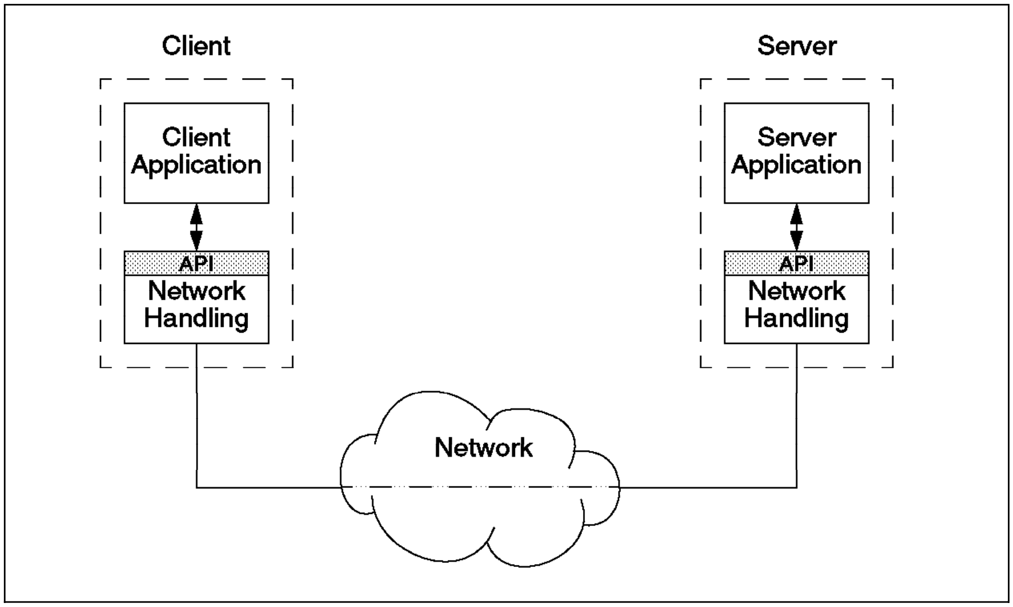


Figure 31. Traditional Communication

Traditional RPC, or even TCP/IP sockets or streams, require the server and clients to incorporate their own security schemes. This includes algorithms to authenticate each other (make sure that each part is actually what it represents itself to be, client or server) and some way to ensure data integrity, not to mention data encryption. Also, a client needs to know, either through customizing or any other service, its server's address in order to establish a connection.

Using DCE RPC as a communication vehicle, as shown in Figure 32 on page 110, transparently involves the DCE Security Service and has several advantages:

- DCE takes care of the authentication process. Servers and clients can trust each other's identity.
- DCE can guarantee that packets sent over the network have not been altered on their way.
- DCE will never transmit any passwords for authentication in plain text form.
- DCE Security Services can be used to encrypt the data sent over the network.

Furthermore, if DCE Directory Service is used as well, clients do not need to be configured in order to find their servers. They can easily look into DCE's directory for their server's address.

Because DCE carries out these functions, application servers and clients can be much easier to implement.

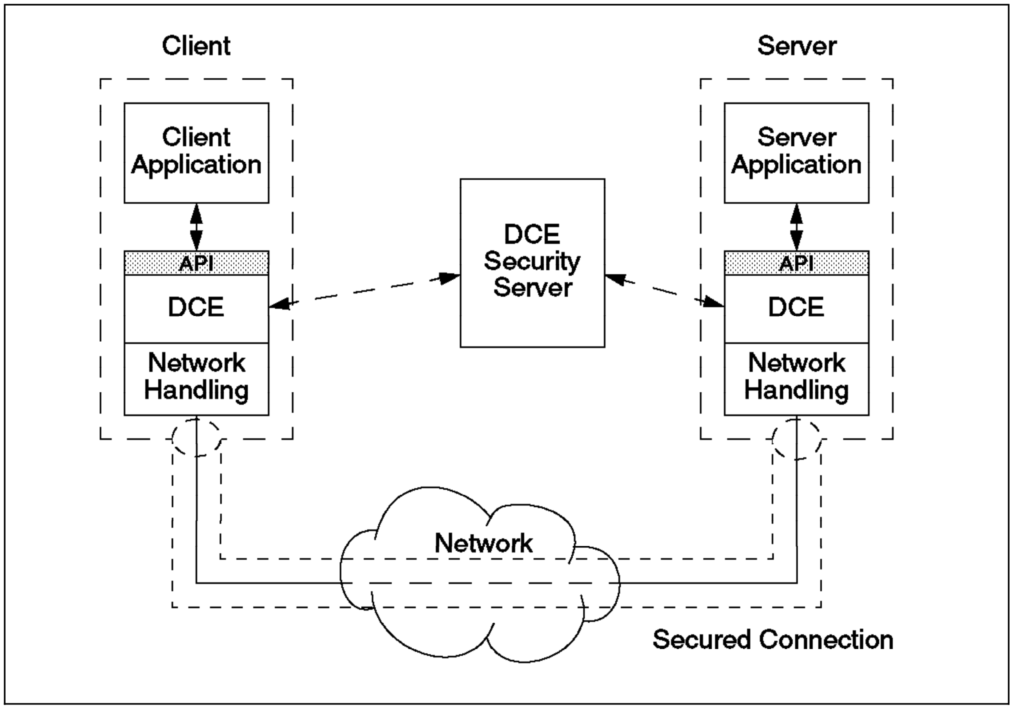


Figure 32. DCE Communication Involving DCE Security Server

7.2 The DCE Programming Model

Programming with DCE can be generally divided up into two different levels, low-level and high-level programming.

7.2.1 Low-Level DCE Programming

1. By use of Remote Procedure Call (RPC), the programmer divides a program into well-defined procedures that are called from the main part of the program. The idea with RPC is that the procedures are then moved to a server system and made available via a DCE runtime environment on the server system. The client can now call the remote-procedures on the server system via its DCE run time. The communication system and protocols are fully transparent for the programmer, and all the interaction between the client and the server is handled by the DCE run time on both systems. DCE's RPC does not require the programmer to know any communication-specific APIs. DCE RPC provides all the functionality that is necessary to hide communication systems and protocols but at the same time provide a true client/server model. OSF DCE Version 1.2 has been extended to support C++ object features, such as inheritance and object references.

2. DCE provides a number of API calls to allow a programmer to use DCE services like Directory, Security, Audit and Management. In general, only

very few API calls are needed to implement a general client/server application that uses standard Directory and Security Services. For use in very specialized application, for example for DCE management applications, DCE provides many advanced API calls, but generally they are not needed.

7.2.2 High-Level DCE Programming

1. An alternative to using the standard DCE programming model is to use higher-level DCE API, which makes the use of DCE a lot easier and speeds up development. Such APIs offer a minimum number of calls and do a lot of the more complex operations implicitly by using default values or making useful assumptions. These calls are easy to use and powerful, but you give up some of the flexibility of the low-level interfaces. High-level programming interfaces are provided by additional tools and products, often called toolkits. There are several toolkits available. Most of them are traditional procedural APIs, such as the EzDCE convenience library described in 7.5.4, "Additional IBM Tools" on page 125.

2. Eventually there are solutions, high-level APIs, where the distributed application programmer does not have to care about DCE. The toolkit is able to use DCE, but can also use another distribution service or even local libraries, whatever is best suited to the environment the application is to run in. Examples are database systems or MQSeries. For example, database systems use DCE to locate services, and communication between remote objects may be based on secure RPC. In MQSeries, the DCE Directory Service can be used to locate queues, and the DCE Security Service for access control or authentication. Similarly, there are toolkits that allow existing client/server applications using another communication paradigm to implicitly use DCE by tunneling the other communication protocol into DCE RPC.

The following sections of this chapter describe application development tools and techniques grouped by these different approaches and discuss some aspects of converting existing applications, so-called legacy applications, into DCE applications.

7.3 DCE Application Development: What Does It Mean?

Normal DCE development means using mainly C language and DCE standard APIs as well as tools to define RPCs and to interface with the Security and Directory Servers. It may also mean using C++ and DCE-wrapper class libraries that provide a low-level or one-to-one mapping of DCE calls to object methods. This way of application development has the reputation of being difficult, but this is only partially true.

As mentioned in section 7.2, "The DCE Programming Model" on page 110, core RPC programming does not involve any special programming effort and does not involve the use of special APIs. It is only when DCE services are going to be used in the program that some special DCE programming facilities and API are going to be used.

OSF DCE offers five core services. They are RPC, Threads, Security, Directory and Time Services. These services are, from a programmer's point of view, described in the following sections.

7.3.1 The RPC Paradigm

The RPC paradigm used in DCE programming allows a programmer to write distributed applications very easily. In many cases, it is possible to write a classic monolithic application, test it, and then split it into client and server parts. The functional parts of the program are not modified by the use of DCE. Using a remote procedure in the client or implementing it in the server is the same as if it were local. It is much easier for a C, COBOL, or FORTRAN programmer with no communication experience to use RPC rather than to use other client/server paradigms, such as programming direct to the TCP/IP sockets interface or APPC or to switch to object-oriented programming.

7.3.2 Using Threads

Thread programming is not mandatory, neither for the client nor for the server side. The server side will automatically use threads to support concurrent connection from multiple clients. Clients will only use threads in special cases, exploiting their advantages.

Using threads allows servers to handle several clients' calls at the same time without the overhead of having many processes. In this case, their use is transparent to the DCE application programmer. But the programmer must be aware that there are constraints:

- The stack size is limited.
- Writes to global objects cannot be used without using mutexes as a lock mechanism.
- Non-thread-safe libraries or APIs cannot be used without using mutexes.

On the client side, threads can also be used explicitly to give more flexibility and power to the programmer. For instance, threads can be used to provide asynchronous RPCs to do background processing while interacting with the user and so on. In this case, the programmer must learn more about their use and about the threads API.

The implicit or explicit use of threads appears to be easy to learn for programmers. Some programmers may already be familiar with it because threads are now implemented in many operating systems, such as OS/2. But if threads imply too many technical or cultural problems, you can still program DCE servers without using them. In this case, you will have to be conscious about the performance issues.

7.3.3 Using Security Services

Security is very important in a distributed client/server application. The server must trust the client. But it is also important to notice that computer server systems are not always located in a physically secure environment. Because of this fact, the client can not trust the server to be the right server. DCE Security Services provide the mechanisms to ensure that the server can trust the client and the client can trust the server. This is called DCE Authentication.

The DCE Authentication Service is easy to program in a client/server application; only few API calls are required.

The Basic Security: DCE Authentication: The simplest level of security is to ask for the server and the client to be authenticated and use a secured level of RPCs. Doing this offers a good level of security:

- The client is sure that the server application is the one it wants and not a Trojan horse.
- The server can check the client's identity and compare it against a list of authorized users.
- The network conversation between the client and the server can be protected by using secured check sums and encryption facilities.

The cost for such security is reasonable. The typical code that needs to be written is:

- Approximately 20 to 30 lines of C code in the client. This includes the use of seven DCE API calls.
- Approximately 75 to 100 lines of C code in the server. This includes roughly 20 DCE API calls. This includes the use of threads that are needed to manage key and login context.

These numbers are based on the Employee Database Application example program.

Sample code is widely available, but the use of security implies an additional administration cost. You will have to create principals for the servers, create their keys, and so forth. It is possible to write portable and reusable scripts for this.

Improved Security: DCE Authorization: DCE Authentication provides the service that enables clients to trust servers and servers to trust clients. In addition, protection of the communication between the client and the server is protected.

The next level of protection is to verify if a specific client can get access to a specific object, for example a record in a database, and what access rights the client has to the specific object.

DCE Authorization provides this functionality. An Access Control List (ACL) manager must be attached to the objects that are to be protected. An ACL manager, in DCE terms, has a number of well-defined functions that must be implemented to ensure proper functionality. One of the functions an ACL manager must provide is remote administration of the ACL.

You have to write an ACL Manager integrated into your application to implement an authorization policy. An ACL Manager is a program that needs to create code underneath predefined ACL interface routines, administer a permission set for the objects it protects, and perform authorization checks.

In earlier versions of OSF DCE, it was very complicated to write ACL managers. Although samples are available, they may need a lot of modifications to meet your needs. So usually, an ACL manager is 2000 to 4000 lines of C code composed of DCE API calls and ACL storage management. This is one of the most complex parts of DCE programming.

OSF DCE Version 1.1 provides a much better programming environment for implementing an ACL manager. The code required to implement an ACL manager has been reduced remarkably. Nevertheless, implementing an ACL Manager is a complex part of DCE programming.

7.3.4 Using Directory Services

Using the Directory Server can be avoided, but this is one of the most important features of a distributed environment. However, this is where the use of DCE APIs become somewhat more complicated. A server should always export its interfaces into CDS when it starts up and remove them from CDS upon termination. This prevents clients from using binding information to servers that are not active anymore and leads to unnecessary time-outs.

For a small program exporting a simple RPC interface to the Directory Server without using multiple objects and different managers, the typical code that needs to be written is:

- From 30 to 100 lines of C in the server. This is mainly 5 to 10 calls to the API in order to register the name of the server and export the interface.
- From 10 to 70 lines of C in the client. This is mainly 5 to 10 calls to the API in order to find a server by its name and bind to it.

If you want to do more complex operations, like exporting multiple objects, and have your server called when they are accessed or allow server replication and automatic rebinding of the client in case of failure, you will have to write much more code, consisting mainly of calls to the API with a lot of parameters and complex structures.

But the good news is:

- Samples for such code are widely available. They are often given with DCE implementations and in books about DCE programming.
- This code is nearly always the same from application to application. So you can pick up a sample, modify it slightly to meet your exact needs, and then use it over and over again.

In fact, understanding the CDS concepts and how the various API calls relate to them is the difficult part of the job. This may take weeks, but when this is done, it is possible to write this part of an application in a few hours.

7.3.5 Using Time Services

Although the DCE Distributed Time Service offers a number of API calls, most applications will not use the DCE Time Service API calls. The DCE Time Service API calls are mainly used by system programmers and by the DCE core services themselves.

7.3.6 How to Create DCE-Based Applications

As long as they are aware of the thread usage implications, especially in the server part, programmers can continue to write the functional parts with little knowledge of DCE. In many cases, they can use their usual language as long as it can be linked with C.

It is very important, then, to have at least one person or, even better, a small team that has good skills and experience in DCE programming. Good candidates are those with knowledge in C and client/server. Their tasks are to find and adapt, or write and maintain, the DCE parts of the applications and to assist the other programmers. For each application, they will at least have to:

- Write the IDL corresponding to the interface between the client and the server

- Write the DCE part of the application that uses the Directory Server.
- Write the DCE part of the application that implements security.

Acquiring these skills can be as short as one month for a well-trained C programmer with experience in distributed systems, non-DCE RPC, and extensive use of books and samples. But it will probably often take two to six months.

7.4 Converting Legacy Applications

Adding DCE to an existing application allows you to integrate them into a client/server information system with consistent administration and at the same time use Directory and Security Services.

Existing applications can be divided into three main categories:

1. Server- or mainframe-based applications. These applications are only accessible via terminal emulators. The reason to convert such applications is to integrate them into, for example, other PC-based applications like workflow, e-mail, and document-management systems. Both users and administrators benefit from using the DCE Security and Directory Services, and the users will at the same time get a consistent user interface. Since there is no specific communication system and protocol in place, the best way to convert these applications is to use RPCs. See section 7.4.1, “Using RPCs,” for more information.
2. Client/server-based applications, where the existing communication system and protocols can be replaced by RPCs. This can be accomplished if the applications have a well-defined interface to the communication system and protocols. See section 7.4.1, “Using RPCs,” for more information.
3. Client/Server-based application, where the existing communication system and protocols cannot be easily replaced by RPCs. If the applications are tightly bound to the communication system and protocols and RPCs can not replace it, then use the Generic Security Service API offered by OSF DCE. GSS-API offers OSF DCE security facilities to non-RPC-based applications. See section 7.4.2, “Using the Generic Security Service API” on page 116, for more information.

7.4.1 Using RPCs

Each application is a special case, but there are some rules that can be used to find the best solution and evaluate the complexity of the task. An application that already has a client/server structure is easier to convert to DCE than a monolithic application. The difficulty of converting a monolithic application is finding the best cut-off line to split it into a client and a server part. The least you can do for the client is to isolate the terminal I/O from the application. This gives you flexibility on both ends; the client can get a user-friendly GUI, and the server can be migrated and even can change its implementation.

We can distinguish two categories of applications that need a different approach:

- If the application already works in client/server mode, the amount of work depends on the communication paradigm used and the quality of the code. Applications using another RPC mechanism or conversational mode (LU 6.2, sockets) are relatively easy to convert. Message queuing applications can make the job more difficult, especially if the communication part is not well separated from the functional part. If the server part is not already thread-safe, it may be very difficult to change. In this case, it is better to run that server single-threaded with a performance penalty.

- If the application is not in client/server mode but can be modified, it may be easy to split it up into client and server parts. This depends on the modularity and the documentation of the code. If it is made of separate procedures or modules, with no or few global variables, it is easy to convert. If not, it is probably better to use another method.

In both cases, the idea is to isolate the server part of the application into a number of well-defined procedures. These procedures and their arguments can now be described in a DCE contract, also called an IDL file. The IDL file is used to generate the DCE communication code that allows the client and server to communicate with each other. The generation of communication code, also called stubs, is done automatically by using the IDL compiler.

The next step is to write the interface between the client code and the RPCs as well as the interface between the server code and RPCs. Also the code to initialize the runtime, Directory, and Security Service must be written, both for the client and for the server part. See also section 7.3.4, “Using Directory Services” on page 114 and section 7.3.3, “Using Security Services” on page 112.

If your application has its own security mechanism, moving to DCE Security may be difficult. Implementing DCE Authorization with ACL manager can be especially difficult. The solution can be to just use DCE Authentication and in that way benefit with single login, simplified user administration, and protection of the communication between the client and the server.

There are tools on the market that make the conversion of legacy applications into DCE applications an easy job. An example is in section 7.5.1.1, “IBM OS/390 Application Servers” on page 119.

7.4.2 Using the Generic Security Service API

The Generic Security Service API (GSS-API) provides an alternative way of securing distributed applications that handle network communications by themselves. With the services offered by GSS, applications can establish secure connections and get the same security facilities that DCE RPC-based applications have.

The GSS-API is standardized by the Internet Engineering Task Force (IETF) and is described in RFC 1508 and 1509. As shown in Figure 33 on page 117, applications call the GSS-API, and the underlying security service is, in theory, pluggable.

Existing client/server applications that use their own protocol for communication, for example TCP/IP socket or advanced program-to-program communication (APPC), can continue to use their own communication mechanism but at the same time use DCE Security Service (see Figure 34 on page 117).

The GSS-API was introduced with OSF DCE Version 1.1.

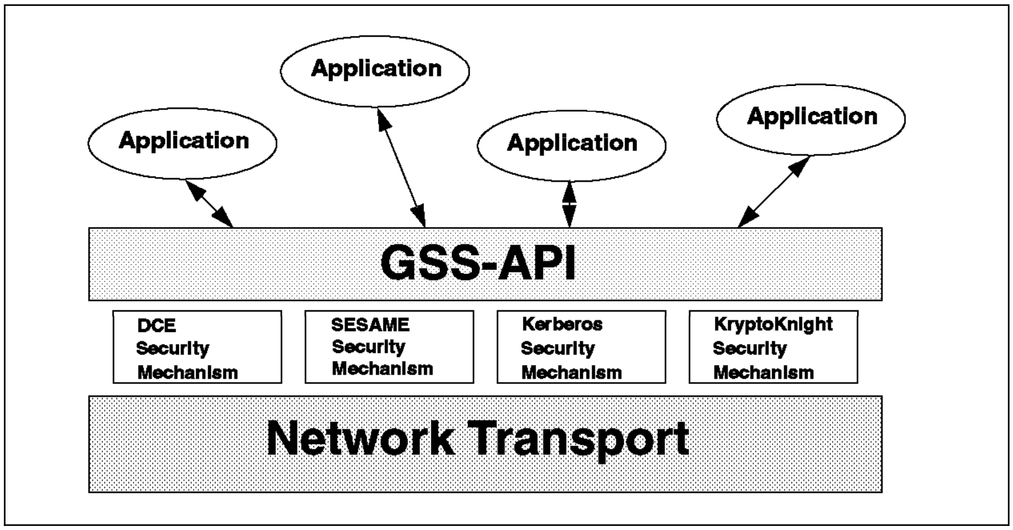


Figure 33. GSS-API Is an Open Standard

IBM has implemented GSS-API on OS/2, AIX, OS/390, and Windows NT. AS/400 and VM are expected to follow. Since GSS-API is a part of OSF DCE Version 1.1, all vendors that implement this version are expected to offer GSS-API on their platforms.

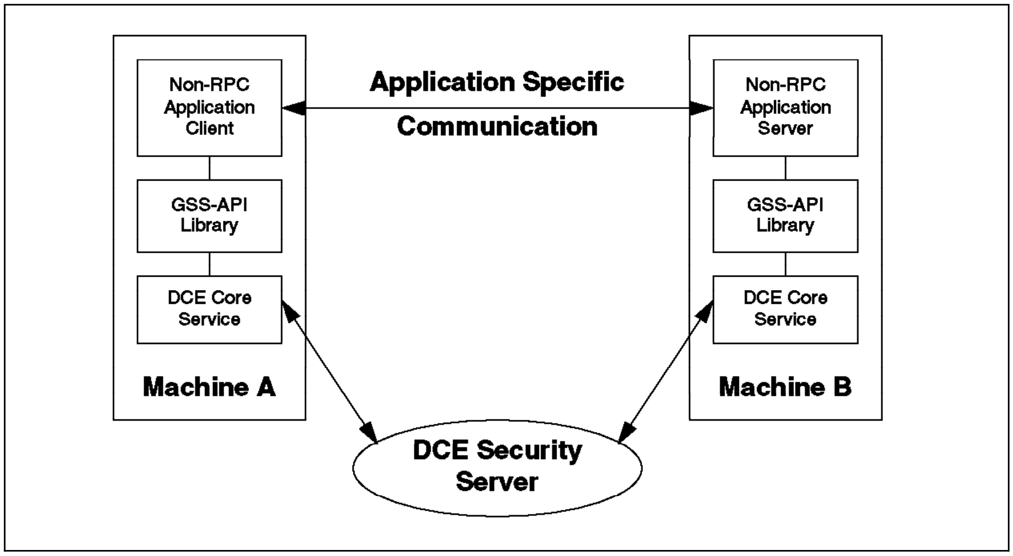


Figure 34. OSF DCE Generic Security Service API

The primary goal of GSS-API in DCE is to provide non-RPC applications access to the following DCE Security Services:

- Authentication: Allows client applications and remote resource managers to authenticate one another
- Security Context Management: Allows users and remote resource managers to build secure connections on top of insecure transport or association services
- Message Integrity and Protection: Allows applications to protect the integrity and privacy of data that flows in the distributed environment

A GSS-API caller (usually the application client) accepts tokens provided to it by its local GSS-API implementation and transfers the tokens to a peer (usually the application server) on a remote system. That peer then passes the received tokens to its local GSS-API implementation for processing.

Clients as well as servers first have to authenticate themselves with the Security Server (network login). Clients then use the `gss_init_sec_context()` call to create the token needed to access a server. In the DCE environment, this call results in a request to the privilege service for a service ticket to the specified server, presenting the client TGT (ticket granting ticket) obtained from the login. This ticket contains the DCE Extended Privilege Attribute Certificate (EPAC) and is returned to the client within the GSS-API security context token. The client sends the token to the server, which hands it over to the GSS-API. The server-side GSS-API verifies the client security context and keeps it for later references until it is destroyed.

In addition to exchanging tokens for initial authentication, the application (client or server) can then create and validate tokens, so-called signature tokens, to protect messages. The sender of a message requests a signature token from the GSS-API and sends the token together with the message. The receiver calls the GSS-API to verify the signature token along with the message.

A typical GSS-API caller is itself a communications protocol that calls on the GSS-API in order to protect its communications with the underlying security mechanisms.

7.5 DCE Development Solutions on the Market

Many development tools allow you to use DCE mechanisms in your applications without having to deal with the complexity of the DCE APIs. Some of them try to simplify the process of standard DCE development or to give more flexibility, and most of them provide a full development environment and paradigm and use DCE underneath to provide services, portability, and interoperability.

Since the market is growing and changing rapidly, it is not possible to describe all the tools available. Many solutions are working on a limited number of platforms or being developed or adapted to DCE. So, we will concentrate on currently available solutions.

Many or all of these products are listed in OSFs product catalog mentioned in A.3, “The Open Software Registry” on page 151, or A.4, “The OSF DCE Product Catalog” on page 151. There you will also find the vendors’ addresses.

The information provided here about third-party products is compiled from publicly available sources, such as the vendor’s Web pages on the Internet.

This section is divided into four subsections:

1. 7.5.1, “DCE Development Tools” on page 119 assists the programmer in the development of DCE client and server applications by discussing a number of different tools and code generators.
2. 7.5.2, “DCE Object Oriented Development Tools” on page 122 discusses a variety of object technology tools. OSF DCE Version 1.2 supports C++ features such as inheritance and object references. A number of additional products are available to support the programmer in development of DCE solutions by using object technology.

3. 7.5.3, “Middleware that Exploits DCE” on page 123 discusses middleware for use in developing DCE applications. Middleware products allow you to develop distributed applications without having to deal with low-level DCE programming. They use DCE transparently, behind the scenes, to obtain global Directory and Security Services. Because middleware completely hides DCE, the applications can run in any supported environment, not just with the DCE. This can greatly increase the programmer′s productivity.
4. 7.5.4, “Additional IBM Tools” on page 125 discusses a number of IBM development and debugging tools. These tools are available internally for IBM employees via tools disks and for customers via the *Developer Connection CD-ROM* (DEVCON) offering. For further information, please see A.9, “The Developer Connection CD-ROM” on page 153.

7.5.1 DCE Development Tools

7.5.1.1 IBM OS/390 Application Servers

DCE Application Support, which runs on OS/390, bridges DCE to the traditional world of OS/390. There are other products on the market today that allow PC clients to access CICS or IMS data and in some cases, their transaction logic as well. These products involve communications gateways that convert other network protocols into SNA in order to present CICS and IMS with familiar requests. DCE Application Support uses TCP/IP.

DCE Application Support allows current CICS and IMS transactions to be called from a DCE RPC application that also utilizes the other DCE services, such as Directory and Security, provided in OS/390.

DCE Application Support allows PC users to access CICS and IMS without communication gateway products by using standard DCE RPC.

DCE Application Support is designed to maximize the current investment in COBOL skills and transactions. A structured transaction can be made available as a DCE server in very short order. Supported transactions require no conversion. New CICS and IMS servers can be written in COBOL transactions. The client code is written in C with DCE RPC, or a development tool can be used. The server handles the conversion of C and COBOL data-types automatically. Finally, the server can coexist with all other ways of accessing CICS or IMS.

7.5.1.2 IBM VisualAge Generator

VisualAge Generator is a powerful, high-end application development solution for building mission-critical, multitier client/server systems. Its object-oriented visual development environment improves productivity, and its platform-independent fourth-generation language shields the programmer from the complexity of the execution systems.

From a single desktop environment, the programmer can rapidly develop and test both client and server logic. The tight integration of the visual development and test facilities results in a true iterative Rapid Application Development (RAD) environment. Once the system is completely tested, the programmer transforms the logic, by using a tool, into an optimized 3GL (COBOL or C++) and deploys it to the target systems.

VisualAge Generator provides support for DCE as the communication mechanisms between client and server.

VisualAge Generator supports many different platforms, including OS/2, AIX, OS/390, OS/400, VM, Windows 3.11, Windows 95, and Windows NT.

7.5.1.3 IBM APPC DCE Toolkit

Developed for programmers with System Network Architecture (SNA) APPC, and Common Programming Interface for communication (CPI-C) skills, the APPC DCE Toolkit provides an SNA programming interface, but uses DCE services underneath. This allows applications written to either interface, APPC or CPI-C, to take advantage of DCE Security and Directory Services without the need for rewriting and adapting them for the DCE APIs.

The APPC DCE Toolkit addresses the following areas:

- Password encryption: Current APPC and CPI-C send passwords in clear form over networks. DCE never does this and thus improves the security of APPC or CPI-C programs.
- Directory services: APPC DCE uses DCE Directory Services to find conversation partners.
- Password administration: Passwords are maintained only in the DCE Registry for both, SNA and DCE applications.
- Single Sign-On: As APPC DCE merges SNA and DCE applications on one common services database, it serves as an ideal basis for single sign-on implementations.
- Management: Overall management and administration of directory and security objects are simplified because they are combined in a single subsystem.

APPC DCE is available for the AIX and OS/2 Warp platforms.

Further information and documentation, including code download, can be obtained from the Web URL `http://www.raleigh.ibm.com/adt/adtprod.html`.

7.5.1.4 IBM LN-DCE

LN-DCE is a Lotus Notes and DCE integration tool that provides a development tool, a runtime, and a library distribution tool.

Basically, LN-DCE builds a Lotus Notes interface module based on a DCE IDL file. This module allows a Lotus Notes programmer to write Lotus Notes applications that communicate with DCE server applications. The programming skills required by the Lotus Notes programmer are limited to the LotusScript programming language.

LN-DCE uses the DCE Directory Service to look up servers and uses DCE Security to provide authenticated RPC.

LN-DCE is divided into three parts:

1. A dialog-based development system
2. A project database and maintenance system keeping track of projects' development and their status. The database tracks different versions of

projects according to the DCE standard for version control that is inherited from the IDL file.

3. A library and source code template maintenance system.

LN-DCE is available for IBM internal evaluation. Platform support is OS/2, Windows 95, and Windows NT. IBM employees can download LN-DCE from IBM's Intranet at the following address:
`http://w3.portsmouth.uk.ibm.com/go/client_server/denmark/ln-dce.htm`

It is planned that LN-DCE will be made publicly available on a future release of the IBM Developer Connection CD-ROM (see also A.9, "The Developer Connection CD-ROM" on page 153) and on a tools disk for IBMers to easily download the tool.

7.5.1.5 Entera

Entera, from Open Environment Corporation, is an enterprise operating environment that includes a suite of client/server development tools. Entera supports the use of DCE Security, Directory, RPC and Thread Services in the development of distributed client/server applications.

Entera allows the programmer to choose among many different development tools on the market and lets the programmer concentrate on the business, presentation, and data access logic. Entera then generates the code needed to build a client/server application, and this includes the use of DCE services.

Entera has native support for many different platforms, including OS/2, Windows, Windows NT, OpenVMS, and various UNIX implementations. Entera also supports integration to MVS and MVS Open Edition. Entera supports many different development tools, among these are VisualAge C++, Visual Basic, and PowerBuilder. Entera supports data access to many different databases, including DB/2, Oracle, Informix, and Sybase.

7.5.1.6 Visual-DCE

Visual-DCE is a product from Gradient Technologies Inc. that offers DCE capabilities to Visual Basic. Visual-DCE provides the client programmer with a number of tools, also called custom controls that allow a programmer who has no C programming or DCE API skills to program the client.

Visual-DCE provides custom controls that represent DCE bindings, interfaces, names, and security contexts. The programmers can create and modify them dynamically. The IDL compiler is extended to handle Visual Basic data-types.

Visual-DCE supports Windows 3.10 and 3.11.

7.5.1.7 Doxa Distribution Tool Kit

The Doxa Distribution Tool Kit (DDTK), made by Doxa Informatique, is a collection of middleware tools running on top of DCE, using Threads, RPC, Directory Service, and Security (optional). It offers a simple interface for writing distributed applications (15 primitives). Key features include:

- An interface compiler based on CORBA IDL
- C or COBOL interface generation and ASN.1 encoding or decoding tables
- A message queue mechanism, allowing blocking and non-blocking, mono- or bi-directional communication

- Efficient use of server computing resources with a reduced number of processes and low context-switching overhead
- Optimization for low-speed lines by giving, for instance, the size of each message during IDL compilation
- Management support with an agent built into the DDTK kernel and management tool provided

The Doxa Distribution Tool Kit is available for AIX, OS/2, and SCO. Planned versions include Windows 3.1, HP/UX, and Solaris.

7.5.1.8 MICRO FOCUS COBOL

MICRO FOCUS COBOL, from MICRO FOCUS Ltd., facilitates the development of COBOL/DCE applications or migration of existing COBOL applications to DCE by providing a COBOL IDL compiler and access to DCE APIs from COBOL.

7.5.1.9 Ellery Open Systems

Ellery Open Systems, from Ellery Systems Inc., provides a complete set of development tools for DCE applications consisting of a C interpreter, debugger, performance profiler, Motif UI builder, a set of Motif widgets for simplifying DCE administration services, and many sample applications

Ellery Open Systems supports AIX and various other UNIX platforms.

7.5.2 DCE Object Oriented Development Tools

7.5.2.1 ANSAware

ANSAware, from APM Ltd., is a set of tools and supporting services for building distributed object applications in C. It provides a easy-to-use veneer over DCE Threads and RPC for the application programmer, together with a set of services to support distributed application management

Platform support: MVS, SunOS, HP-UX, MS-DOS, OpenVMS, and Windows 3.1.

7.5.2.2 Distributed ObjectIQ

Distributed ObjectIQ, from Hitachi Ltd., is an add-on to the ObjectIQ product, also a product from Hitachi Ltd., that allows the programmer to construct distributed applications quickly and simply.

ObjectIQ is a general-purpose application development tool that is based on an object-oriented and rule-based programming language. Distributed ObjectIQ automatically generates the code needed to support the underlying DCE services, like DCE Security.

ObjectIQ supports many different UNIX systems, including IBM AIX.

7.5.2.3 SNAP

SNAP, from Template Software Inc., is an advanced OO development environment that provides reusable software components for development of complex distributed applications.

SNAP is available on UNIX, NT, and OpenVMS.

7.5.2.4 COHESIONworX

COHESIONworX, from Digital Equipment Corporation, is a distributed software development environment and framework for UNIX. It is based on an object-based desktop and contains a complete set of programming tools for the code-edit-debug phases of development.

Based on DCE, the COHESIONworX development environment provides an integration framework that links distributed systems to form a single virtual system on each developer′s desktop.

COHESIONworX is available on HP, SUN, and Digital UNIX.

7.5.2.5 Vendor Specific Development Tools

Some vendors offer object oriented application development tools for distributed DCE applications on their specific platforms.

7.5.3 Middleware that Exploits DCE

7.5.3.1 IBM MQSeries

IBM MQSeries is a message queuing interface that is available for many IBM and non-IBM platforms. It provides for simple asynchronous (independently operating, non-blocking) communication, using queues. The messaging and queuing support routes the message to the appropriate destination in the network so that it can be accessed by the programs servicing the queue. The current AIX version allows the use of the DCE Directory Server to locate queue servers. By doing this, a client application can connect to a queue by using its name. The location of the queue server can change without having to change the configuration of the client machine. This is totally transparent for the applications and for the programmers.

7.5.3.2 Encina

Encina, by Transarc Corporation, provides transaction facilities to DCE. While DCE addresses many of the complexities of distributed computing, developers of distributed applications frequently face additional requirements for which the transaction processing paradigm is important. Transaction processing is fundamental to enterprise computing because it provides data integrity in the face of concurrence and system failures. In the Encina family of products for open, distributed transaction processing, Transarc has extended DCE to simplify the construction of reliable distributed systems and to provide the integrity guarantees required for mission-critical enterprise computing.

Encina provides the tools that allow the programmer to build transaction-processing applications based on DCE services. Encina also serves as an excellent application development tool for DCE applications, where transaction-processing is not a major requirement.

Many platforms are supported both as client and server. Encina for AIX is part of the *IBM Transaction Server for AIX* product, which also includes IBM CICS for AIX. Encina for Windows NT is part of the the *IBM Transaction Server for Windows NT* product, which also includes IBM CICS for Windows NT (see also 6.5, “IBM Transaction Servers” on page 95 for further details).

7.5.3.3 IBM CICS

IBM CICS for AIX is part of the *IBM Transaction Server for AIX* product, together with Transarc's Encina for AIX, or IBM CICS for Windows NT is part of the *IBM Transaction Server for Windows NT* product (see also 6.5, "IBM Transaction Servers" on page 95). CICS is also available from IBM for the Sun Solaris operating system, making use of Encina for Solaris and DCE services the same way as on AIX and Windows NT.

IBM CICS offers an interface for CICS-based applications on the AIX and Windows NT platform. Customers can protect their investment in CICS applications and skills and they can rightsize their transaction-oriented CICS applications by including AIX and Windows NT systems.

The Transaction Servers for AIX and Windows NT include AIX and Windows NT clients for Encina and CICS, the new IBM CICS clients for OS/2, DOS, Windows+, and Macintosh as well as the CICS Client for Sun Systems. The new CICS gateway provides access to existing CICS applications for Lotus Notes and the Internet.

7.5.3.4 Open TP1

Open TP1, by Hitachi Ltd., is an infrastructure for mission-critical, distributed online transaction processing systems. The X/Open distributed transaction processing model is used as reference for the development of Open TP1.

Open TP1 supports DCE services and provides a consistent environment for developing and using distributed applications.

Open TP1 supports IBM AIX and many other platforms.

7.5.3.5 Connection/Encina

Connection/Encina, from Open Horizon Inc., is designed to be used in a DCE-based three-tier architecture where Encina from Transarc acts as a transaction manager. Each client transaction may result in several database interactions that need to be committed as a single transaction. Connection/Encina supports this.

Connection/Encina interfaces between the Encina transaction system and the database systems and supports all database systems that offer the Open Database Connectivity (ODBC) architecture.

Connection/Encina supports the following platforms: IBM AIX, MVS, Windows 3.1, Solaris 2.3, HP/UX, and OpenVMS.

7.5.3.6 Connection/DCE

Connection/DCE, by Open Horizon Ltd., is a database connectivity product that offers full DCE integration. Connection/DCE transforms ODBC calls into authenticated RPC calls and supports DCE Security and Directory Services.

Connection/DCE maps the users DCE login to the database login to provide single sign-on. Since ODBCs are replaced with authenticated RPC calls, the users have the option to protect the data when it passes over the network.

Connection/DCE is fully transparent for the applications because Connection/DCE provides a native ODBC interface.

Connection/DCE supports the following platforms: IBM AIX, MVS, Windows, Solaris, HP/UX, and OpenVMS

7.5.3.7 DCE/Snare

DCE/Snare from IntelliSoft Corp. is a DCE-based tool that intercepts standard TCP/IP communication requests from applications and adds DCE Security to them by a tunneling function through DCE RPC or GSS-API secured communication. Centralized configuration and administration is provided. DCE/Snare is suitable for existing legacy applications or for new applications. Support for light clients, which are not DCE clients, is provided by DCE/Snare-Light.

DCE/Snare is available for several platforms.

7.5.3.8 Oracle7 Advanced Networking Option

Oracle Corporation supports DCE. Oracle Tools and CASE products can use the Advanced Networking Option (formerly known as SQL*Net/DCE) as the mechanism for communication between clients and servers.

The Advanced Networking Option uses authenticated RPC for communication between client and server. It also uses DCE Directory and Security Services.

7.5.3.9 Sybase Open Client and Open Server

Open Client and Open Server, from Sybase, Inc., is the foundation for all Sybase Enterprise Connection products.

Open Client and Open Server supports DCE Security, Directory, and Thread Services.

For platform support, contact Sybase.

7.5.4 Additional IBM Tools

Some DCE-related tools have been developed at IBM. Although they are not IBM products with commercial and technical support, they may be very useful. Customers can obtain these tools from the *IBM Developer Connection CD-ROM*. For more information, please see A.9, “The Developer Connection CD-ROM” on page 153.

7.5.4.1 Distributed Application Development Toolkit for OS/2

The Distributed Application Development Toolkit for OS/2 is a new set of VisualAge C++ tools that enables you to design and develop applications using object-oriented technology

The toolkit allows both novice and experienced programmers to produce C++ code in either a visual or nonvisual environment, using either object-oriented or procedural programming techniques. Wherever possible, the toolkit uses default settings to control middleware behavior, which simplifies the development process and minimizes the level of skill required for the novice programmer. However, the toolkit also permits fine tuning for experienced programmers who require a greater degree of control over middleware configuration.

If business or technological reengineering require that you enable your distributed logic application to interact across heterogeneous systems, the toolkit can help you make the required changes. Using the toolkit, you can also scale your application up to the level of very large and complex corporate applications.

The toolkit includes the following features:

- Complete runtime environment
- Simplified coding and concurrent use of multiple middleware technologies, including DCE, MQSeries, and TCP/IP
- Flexible design tool for developing and distributing application logic across clients, agents, and servers
- Powerful code generator that emits client, server, and shared parts to support the DCE, MQSeries, and TCP/IP middleware, including source and header files, distributed interface files, and command scripts
- Distribution Class Library that enables you to add new middleware and write middleware-independent applications
- Distributed event trace facility to view distribution communications and instrumentation APIs to record application-level events
- Integration with the VisualAge C++ compiler, editor WorkFrame, Visual Builder, and Open Class Library

7.5.4.2 The MakeDCE Tool

MakeDCE is a family of productivity tools to speed up the development of DCE client/server applications, relieving the developer from some of the tedious details of DCE. MakeDCE complements the DCE basic tool set and can be used to reengineer existing applications into client/server applications. It is available on OS/2 and AIX platforms.

MakeDCE provides two utilities: idlgen and gluegen:

- The idlgen utilityis instrumental in developing IDL files from C files (existing programs) and for maintaining the affinity between these two types of separate, though parallel, sources.
- The gluegen utility is instrumental in bringing the DCE services into the application by using a small, high-level and object-oriented declarative specification language that frees the programmer from meddling with the intimate details that the DCE runtime supports. It generates glue code that wraps the existing application to establish a DCE working environment around plain C code.

7.5.4.3 The EzDCE Convenience Library

EzDCE is a tool, an API, that provides a high-level abstraction of the DCE APIs. This allows an application developer to write to a smaller, more powerful API than that available natively. This should allow the developer to write DCE client and server code much more effectively and efficiently. It works on AIX 3.2.5.

If you are working for IBM, you can also access this tool from your VM account. To obtain it, type the following command from the VM command line:

```
TOOLS SENDTO AUSVM6 SERTR AIXTOOLS GET DCETOOLS PACKAGE
```

7.5.4.4 The PDDA6000 Debugger

The Parallel and Distributed Dynamic Analyzer is a special version of Tim Bell′s xfdb debugger. It runs under AIXWindows on AIX 3.2.5. It is a powerful tool to debug DCE client/server application. Some of the most interesting features are:

- Debugs clients and servers in the same session
- Provides full threads support with global process or per-thread view
- Can find which client is associated with a thread
- Can set breakpoints in any debugged program, either client or server
- Supports debugging of three-tiered or even more complex architectures
- Includes the features of a normal, non-distributed debugger and can also be used to debug stand-alone programs

If you are working for IBM, you can also access this tool from your VM account. To obtain it, type the following command on the VM command line:

```
TOOLS SENDTO AUSVM6 SERTR AIXTOOLS GET DCETOOLS PACKAGE
```

Chapter 8. DCE Administration Tools

This chapter gives a platform-independent view of DCE administration with standard and non-standard tools. Standard tools are those shipped from OSF together with DCE code; non-standard tools are any other tools on top of DCE, developed from IBM or other vendors. Installation of DCE software and machine configuration are platform specific. Each IBM platform provides its own specific value-added set of tools and integrates them into the regular administration environment. So, DCE administration adheres to the usual semantics of each operating system, and the administrator can work with their familiar tools. For example, AIX offers a suite of shell scripts, such as mkdce, which eventually use OSF commands and integrates them into SMIT. See Chapter 3, "DCE Implementations on IBM Platforms" on page 21 for platform-specific tools.

DCE administration, as far as DCE core services or DFS are concerned, benefits from the DCE client/server design and offers some interesting features:

- The administration can be made from any machine in the cell. This means that you can do administration tasks everywhere, and you can freely choose the administration platform with regard to the quality of tools available on it. From there, all DCE/DFS servers can be managed no matter which platform they run on. Only very few operations may require interventions on a specific machine, usually only during problem determination and resolution.
- Administration rights are granted by ACLs. This means that you can delegate part of the administration to several people. For instance, a local administrator can have the responsibility for some applications and is able to install new versions, to set the access rights for them, and so on. But they will have no other special rights.

Application servers or clients may also provide an administration interface that works in a client/server fashion with the above-mentioned advantages. In the worst case, it may be necessary to configure applications locally on all machines if decent tools are not provided.

8.1 Standard Administration

On most platforms, the DCE implementation makes standard tools that come with DCE from OSF available to the administrator. On OSF DCE 1.1 based implementation, the tool to mention is basically dcecp (DCE Control Program), a single but powerful command that allows to configure DCE, browse information, alter existing configurations, and do almost any kind of administrative tasks on DCE.

Earlier implementations of DCE use a set of different commands for administrative tasks. Note that OSF DCE 1.1 still provides this commands, but they may be discontinued in future releases and should not be used any more. The main commands are:

- rpccp, which is used to administer RPC mappings, profiles, and groups
- cdscp, which is used to administer the Directory Server
- rgy_edit, which is used to administer Security Server objects
- sec_admin, which is used to administer the Security Server

- acl_edit, which is used to set, list modify and remove ACLs
- dtscp, which is used to administer the Time Server
- bak, bos, cm, and fts, which are used to administer DFS

Other tools are available for the GDS administration or other tasks, but they are less common.

The standard DCE administration tools allow you to fully administer DCE. They are portable enough to be found on a wide range of platforms. So, an advanced DCE administrator has to master them in order to be able to perform uncommon tasks or to write his own administration scripts. However, they have some drawbacks:

- Their user interface is poor. They are line-oriented text tools, with no command history or other facilities.
- The user interface is not exactly the same from tool to tool. Some keywords or rules are different.
- In order to perform some tasks, you may have to switch from one tool to the other. For instance, to add a new application server and give access to it, you will have to use rgy_edit, cdscp, and acl_edit.
- The syntax of the commands is sometimes cryptic, and the online help is not complete enough.

The standard tools are neither easy nor comfortable to use. So if you want to administer a production cell with them, you should consider the following:

- Invest in DCE skills. This enables the administrators to master the tools and to write their own scripts for common tasks.
- Use them on a platform that provides a better interface. For instance, the SMIT interface on AIX allows you to do a lot of the DCE administration much more easily.
- Use a platform with good and complete online documentation on DCE and its administration, such as InfoExplorer on AIX.
- Use a platform that lets you create your own administration commands and sequences from the standard commands, such as shell scripts on AIX. An example of this is the set of user management scripts that come with the *Administering IBM DCE and DFS Version 2.1 for AIX (and OS/2 Clients)* redbook.

In OSF DCE 1.1, the single command, dcecp, is extendible by means of the TCL scripting language. It is available on platforms with OSF DCE 1.1 support.

8.2 Platform-Specific Add-Ons to Standard Tools

On top of the standard tools that come with DCE, most platforms provide additional tools that build an easier-to-manage interface to the standard tools. These enhancements are often good enough to manage small DCE cells. For example, AIX uses its Systems Management Interface Tool (SMIT) and allows you to do most administrative work, such configuring DCE servers and clients, add/change/remove users and groups, look at and change parameters, and so forth. Although SMIT is only a command builder that calls standard DCE administration tools, it has important advantages:

- It allows an administrator to select parameters from a list of valid values, where available and applicable. This drastically reduces the likelihood of typing errors and speeds up administration.
- It does some parameter checking before calling the actual command. This reduces the chance of errors due to missing or wrong parameters.
- Through its structure of panels, it guides the administrator to do multiple tasks in the correct sequence.

On the OS/2 platform, Directory and Security Server for OS/2 Warp, Version 4, includes DCE administration in the award-winning GUI of OS/2 Warp Server.

Similar tools and enhancements to DCE standard tools are available on the other platforms.

8.3 IBM DCE Manager for AIX

The *IBM Distributed Computing Environment Manager for AIX* provides a graphical view of one or more DCE cells and the resources in them. DCE Manager is integrated in IBM NetView for AIX and acts as a logical extension to the management capabilities of NetView for AIX. Using a similar representation technique as NetView, using maps and submaps, DCE Manager presents a familiar user interface and can display several different views of a cell. It can be launched from NetView for AIX and makes use of the NetView for AIX online help facility to provide user help information.

DCE Manager is capable of automatically discovering and monitoring all DCE core servers, such as the Security Server, Directory Server, Global Directory Server, Time and RPC, as well as all DFS servers in the cell or cells being managed. DCE Manager acts as a DCE client and can monitor DCE daemons on the systems in the DCE cell. It is capable of managing the new daemons (dced, auditd, pwsyncd, pwstred, time provider) introduced with OSF DCE Version 1.1.

Once the DCE resources have been discovered, either automatically or manually initiated, a polling process is initiated to determine the current status of each DCE server. The current status is displayed in the maps and submaps, allowing a quick overview. Through color-coding, failing components can easily be recognized on such maps.

An important feature of DCE Manager for AIX is its capability to collect dynamic performance statistics from the DCE servers, providing input for a proactive system and performance planning.

For DFS servers, marginal and critical usage thresholds can be set for each aggregate holding filesets. When these thresholds are exceeded, the new status is reflected through a color change of this server on the submaps. Dynamic usage statistics can also be collected from DFS servers, which allow you to plan ahead and to avoid critical situations.

Through its integration with IBM NetView for AIX, DCE Manager for AIX provides excellent management functionality. Since many common DCE problems may be caused by network problems, NetView can help to discover them more efficiently.

For ordering and other additional information, please refer to the AIX section, 3.4.4, "Administration Tools" on page 63, and 3.4.7, "Product Packaging and Prerequisites" on page 67, respectively.

8.4 Tivoli Management Environment, TME 10

With the merger of Tivoli Systems Inc. and IBM Corp. in early 1996, *Tivoli Management Environment 10* (TME 10) became IBM's strategic distributed systems management platform. TME 10 is an open framework that incorporates a broad range of management functions and strategies and allows development of add-on functionality through its open interfaces.

A Tivoli Application Partner, *santix software gmbh*, has announced four modules based on TME 10 that add DCE management to the TME 10 framework. The four modules are:

- **DCEmgmt/Security Manager**

  Extends the basic management of hosts, users, and groups provided by TME 10 for standard UNIX and NIS by adding the according DCE management disciplines for organizations, principals, groups, and accounts. This integrates into a single point of management for user and group administration. A later release of DCEmgmt/Security Manager will support keytab files and extended registry attributes.

- **DCEmgmt/Event Manager**

  Integrates DCE event status management into TME 10's powerful event manager. This allows to monitor distributed server status and collect DCE auditing information.

- **DCEmgmt/Cell Manager**

  This is a basic configuration and administration extension to TME 10 for DCE services. It allows you to configure Security Services, Directory Services, Time Services, and clients, and it manages replication of server services.

- **DCEmgmt/DFS Manager**

  DFS Manager takes care of all DFS-related administration tasks, such as configuration, creation of aggregates and filesets, backup and replication filesets, auditing, and event handling (in conjunction with Event Manager).

At the time of writing this publication, not all of these modules were available. Customers should check with santix software gmbh for further details.

8.5 DCE Cell Manager from Chrisholm Technologies

DCE Cell Manager is a graphical administration tool from Chisholm Technologies Inc. (formerly HaL Software Systems). It consists of the following three Motif-based applications, for which a short description is provided:

- Security Manager administers the security registry. This involves adding, modifying, and deleting organizations, groups, accounts, and principals, and managing group memberships. It also provides for backing up registry databases and managing replicas.
- Namespace Manager combines CDS and RPC functionality to administer the CDS namespace. It allows you to create, move, delete, and link CDS

directories and many other administrative housekeeping tasks in the namespace.

- Configuration Manager provides distributed host-management capabilities, such as starting and stopping DCE services, configuring replication, and checking Time Services. It uses asynchronous SNMP traps to inform any monitor, such as IBM NetView/6000, of DCE daemon failures.

DCE Cell Manager makes DCE administration easy by providing a state-of-the-art graphical interface. Its ability to handle multiple instances of an object at a time makes it suitable for medium and large environments. A host agent can run on all DCE nodes, allowing remote DCE configuration and administration similar to the DCE host daemon, which is part of OSF DCE 1.1.

DCE Cell Manager is available from Chisholm for AIX and other UNIX systems.

8.6 DCE/Sleuth from IntelliSoft Corp.

DCE/Sleuth from IntelliSoft Corp. is a Motif-based graphical tool that allows an administrator to trace DCE-specific traffic on the network. It can be installed on any machine having physical access to the network. Various filtering capabilities provide for selective display and analysis of the DCE-related packages on the network.

DCE/Sleuth can be used for problem determination, capacity planning, and performance analysis and tuning.

8.7 IBM Strategy for DCE Administration

IBM plans to make DCE administration easier by providing powerful tools on its platforms. These tools will have the following characteristics:

- Objects based
- Consistent cross-platform administrative graphical user interface
- SNMP support, thus allowing management from NetView for AIX
- Close integration with other IBM and non-IBM tools

A prototype of a new leading edge administration GUI for DCE on AIX and OS/2 has been shown publicly as a technology demonstration.

Chapter 9. DCE Evolution and Future Directions

This chapter covers future directions of OSF DCE and DCE-based vendor solutions. DCE is still evolving, gaining new features, improving performance, and integrating better into existing information systems.

Although OSF DCE 1.1 was made available by OSF in November 1994, there are still some products on the market and in customer's installations that are based on earlier OSF levels. Because of this, and because OSF DCE 1.1 was a major enhancement release, we first look at the important changes introduced by this release and then explain the new functions introduced by OSF DCE 1.2, which was released by OSF in two steps.

9.1 OSF DCE 1.1

OSF DCE 1.1 was a major release of DCE. It was made available by the OSF at the end of 1994, and resellers soon began to port it to their platforms and integrate it. It is currently available commercially from IBM and other vendors, but some DCE implementations are still based on earlier releases.

Here we discuss the improved features of OSF DCE 1.1. It was developed with a strong emphasis on integration services, enhanced enterprise support, improved administration, a DFS/NFS gateway, and other features.

9.1.1 Integration Services

In DCE, applications can control access based on highly reliable authorization services provided by the DCE Security Service. Other products have their own proprietary mechanisms to protect their client/server environment. The new DCE security features described hereafter allow other products to use DCE security as a pluggable security layer underneath, thus unifying authentication (login) of all these products.

Extended Registry Attributes: Initially designed for UNIX systems, the registry contains a user ID and a group ID that are used to provide DCE principals an identity and credentials to use when accessing services on UNIX systems. Since not all the machines in a distributed environment are running some variant of the UNIX operating system, a mechanism, called Extended Registry Attributes (ERA), for storing user and group attributes for arbitrary operating systems has been added in DCE 1.1.

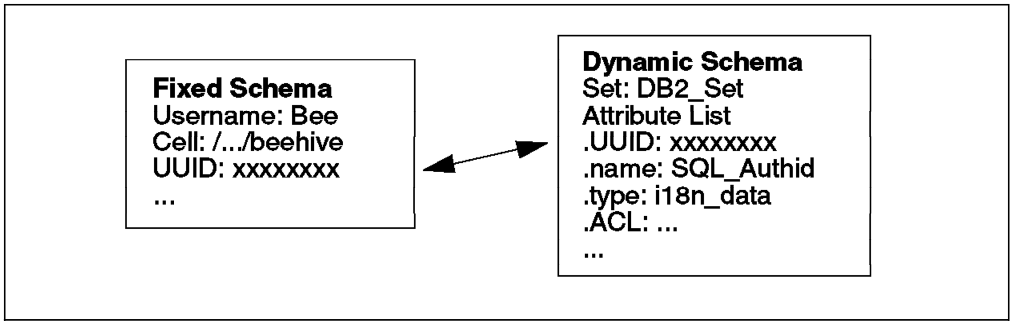


Figure 35. Extended Registry Attributes

DCE 1.1 allows you to define new attributes one at a time. This concept is called the dynamic schema. Figure 35 shows an SQL_Authid attribute definition. An attribute has several definition parameters, such as its name (SQL_Authid), its data type, the option to call another server for information (a trigger function) and more. The trigger allows for an automatic call, for instance, to a legacy registry server when the attribute is accessed upon.

The Extended Registry Attributes (ERA) can be delivered to or retrieved from DCE application servers for authorization checks or further actions as part of extended privilege attribute certificates (EPACs).

Generic Security Service API (GSS-API): The Generic Security Service Application Programming Interface (GSS-API) provides an alternative way of securing distributed applications that handle network communications by themselves. With GSS-API, applications can establish secure connections and act like DCE RPC servers.

The GSS-API is a standard API for interfacing with security services, defined by the IETF RFCs 1508 and 1509. It allows flexible use of the DCE security by programs, even if they don′t use DCE RPC to communicate. Because the GSS-API is product neutral, this allows you to define your security policy and implement it using the generic API.

The GSS-API is explained in more details in 5.1, “Generic Security Service Application Programming Interface (GSS-API)” on page 88.

9.1.2 Enhanced Enterprise Support

The enhancements for enterprise support in OSF DCE 1.1 are:

Hierarchical Cells: With hierarchical cells, independent DCE cells can be connected into a hierarchical cell configuration which makes it easier for companies to build cell structures that correspond to the company structure. Cell names can be registered in CDS, thus making one cell′s CDS the higher-level directory service and the cell itself the parent cell.

Although introduced in OSF DCE Version 1.1, but due to the lack of customer requirements for it, few vendors have implemented hierarchical cells in their products. DSS for AIX, for example, does not support hierarchical cells.

Delegation: Sometimes, as part of the execution of a remote request from a DCE client, a DCE application server needs to issue a request to another DCE server. The last server in the chain needs to authorize the request, but whose permission does the server check, the DCE client′s or the intermediary′s? In Release 1.0.x, the last server in a chain could not see the DCE client. In OSF DCE 1.1, the intermediary server can now pass on the client′s identity without having to change his own identity. The DCE delegation model requires the extension of two components:

1. Extended privilege attribute certificates (EPACs)
2. ACL model

The EPACs need to contain all involved identities. The ACL contains more entry types, such as user_obj_delegate, group_obj_delegate, user_delegate, and so on, to define the permissions of principals whose requests come in via an intermediary. So, the object being protected by ACLs has to set up the according permissions if it wants to allow delegated access.

Note that not all current implementations of OSF DCE 1.1 support delegation.

Extended Login and Password Management: The former DCE 1.0.x login lets anybody try to log in and get some response from the Security Server. Furthermore, the login flow does not let the Security Service know if login succeeds or fails. This has been improved for OSF DCE 1.1. The DCE 1.1 security does not respond to machines it does not know, and it realizes when a user login attempt fails. The new features for login processing are:

- Pre-authentication based on the machine identity
- Account disabling based on invalid attempts

On login attempts, the DCE Security Server only responds if the request comes from a trusted machine and if the user name exists. Messages are encrypted with a strong key based on the machine session key. This is the strongest level of pre-authentication security.

For password management, there are new ERAs (Extended Registry Attributes), for example, for password strength checking, password generation, and password cycles.

Auditing: Administrators can track security-related events within DCE′s trusted computing base.

Audit data structures and a set of audit logging API functions have been defined as well. An API has been provided which can be used by DCE servers and audit-trail analysis tools to log and retrieve audit records. These data-structure and API-function specifications are derived from those of the POSIX 1003.6 standard. For more information, refer to OSF RFCs 28.0, 28.1, 29.0, 29.1, and 29.2.

Internationalization: With OSF DCE 1.1, it is possible to supply message catalogs to localize DCE programs by supplying DCE messages in other languages. This work uses interfaces, as defined in ISO C POSIX 1003.1, 1003.2, and XPG4, including a range of items such as support for multibyte characters, collation, and date and time formatting.

OSF DCE 1.1 also introduces the Character Code Set Interoperability allowing development of RPC applications that automatically convert character data from one code set to another.

9.1.3 Improved Administration

OSF DCE 1.1 introduces new and improved administration functions:

DCE Control Program: The DCE control program (dcecp) available with OSF DCE 1.1 provides consistent, portable, extensive, and secure access to nearly all DCE administration functions from any point in a DCE cell.

This new control command, dcecp, incorporates most of the operations previously performed by using various components′ control programs (rpccp, cdscp, rgy_edit, dtscp, and sec_admin) into a single command line interface. In cooperation with the DCE host daemon, it allows for remote administration of DCE services on other machines.

It also includes TCL (Tool Control Language), a powerful and portable scripting language. TCL is platform independent and runs on every system where DCE Version 1.1 is installed. Portable extensions to dcecp can be written in TCL to simplify and customize administrative tasks.

DCE Host Daemon: The DCE host daemon (dced) replaces the former RPC Daemon (rpcd) and the Security Client Daemon (sec_clientd). And, it enables complete remote administration of DCE services and other applications as well as their configuration parameters.

Cell Name Aliases: Cell aliasing enables cell names to be changed and allows cells to have multiple names to reflect changes in an organization.

Not all implementations actually support cell name aliases. DSS for AIX, for example, does not support it.

Serviceability Improvements: Serviceability improvements enhance the diagnostic messaging capabilities of DCE 1.0.X by instrumenting services to capture more information and unifying the message format across all DCE components. The DCE Control Program provides remote administrative commands to control the generation and routing of messages based upon component and severity. A new document, the *DCE Problem Determination Guide*, provides an explanation and administrative action to be taken for every error code DCE generates.

9.1.4 Performance Enhancements

The IDL compiler generates smaller, cleaner RPC stub code and supports a number of new IDL constructs. The IDL compiler is also a key component for internationalization support.

In addition to this, performance improvements have been made for RPC throughput.

9.1.5 DFS/NFS Gateway

OSF DCE 1.1 introduces a gateway function that bridges DFS into an NFS environment. Users at workstations having only NFS can access DFS files through such a gateway. In order to maintain access security to DFS files, such users will be required to authenticate with the gateway first and will then be granted access to the DFS file tree.

Because of its importance for many customers, IBM has released the DFS to NFS gateway already in the earlier implementation of DCE Version 1.3 for AIX, which was based on OSF DCE 1.0.3.

9.1.6 Other New Features

Other new features include:

- Backing store library for the convenience of programmers who are writing DCE servers. A backing store is a persistent database or persistent object store from which types can be stored and retrieved by a key.
- Group override customizes the group name mapping from host to host to allow DCE to adopt to various operating system conventions.
- ACL Manager library eases development of servers by providing server writers with an ACL manager for use with all servers.

- GDS enhancements provide modifications to various GDS components to improve ease of programming and administration.
- DFS delegation allows a file to be passed with the initiator′s privileges intact.
- Subtree operations allows large-scale administrative name changes within cells.
- Distributed Time Server (DTS) enhancements provide for remote administration of DTS.

9.2 OSF DCE 1.2

This section describes the goals and contents of OSF DCE 1.2. OSF makes this release available in two steps, OSF DCE 1.2.1 and OSF DCE 1.2.2. The first, OSF DCE 1.2.1, was made available by OSF in March 1996, but products cannot typically be expected before one year later. OSF DCE 1.2.2 is planned to be made available in December 1996.

OSF DCE 1.2 is the first project to be operated under OSF′s Pre-Structure Technology model (PST). In this model, the project is planned, managed, developed, and funded by a group of project sponsors. The sponsors of DCE Release 1.2 are IBM, Digital, HP, and Hitachi. These sponsors, a representative of the OSF End User Steering Committee, and a representative of the OSF permanent staff form a Project Steering Committee (PSC) that is responsible for running the program. The PSC has designated IBM as the prime contractor to take responsibility for managing the development, integration, and delivery of DCE 1.2 to OSF.

The primary goal of OSF DCE 1.2 is to facilitate the continuing trend toward enterprise-wide deployment of DCE in the 1997-1998 time frame.

In the following sections, we describe the main new features of OSF DCE 1.2.1 and OSF DCE 1.2.2 by functions.

Not all vendors will, however, implement all features of a new OSF DCE release. IBM will continue its effort to implement those functions that customers really need and ask for and will continue to enhance its DCE-based products and implementations to improve reliability, serviceability and functionality beyond the new features introduced by OSF.

9.2.1 Integration with Other Environments

Few DCE installations are deployed in isolation. Most environments will include a mixture of existing technologies that have some degree of overlap with services provided by DCE. DCE will address this overlap of functionality by allowing access between DCE and the most popular of these environments. DCE will reduce the administrative overhead associated with managing multiple environments by eliminating the maintenance of duplicate data and integrating administrative operations.

- Novell NetWare Coexistence

  OSF DCE 1.2.1 incorporates file sharing services and administrative aids that allow NetWare 3.X users and DCE users to have a single identity and a single view of the DCE file system, DFS. OSF DCE 1.2.1 also unites DCE and NetWare user and group administration by providing a set of utilities and a

NetWare Loadable Module (NLM) that together enable management and synchronization of user accounts across both systems.

- Open Network Computing (ONC) Coexistence

  ONC, especially Network File System (NFS), exists on a large number of machines in many different operating environments. ONC integration enhances the secure DFS/NFS gateway of DCE Release 1.1 with support for the DFS host-specific (@HOST) and architecture-specific (@SYS) file naming features.

9.2.2 Administration Enhancements

OSF DCE 1.2.1 continues to improve DCE administration. The DCE Control Program (dcecp) introduced with OSF DCE 1.1 has been enhanced by adding new functions and extensions. Included are new commands, new actions on existing commands, and new options on existing commands.

These enhancements emphasize OSF's ongoing effort to ease administration of DCE and to concentrate it into a single, universal administration command.

9.2.3 Object Oriented Programming

OSF DCE 1.2.1 provides C++ support to IDL. This support allows client and server programs written in C++ to utilize DCE RPC in a highly transparent manner using natural C++ constructs. The IDL language has been extended to support C++ features such as inheritance and object references. However, the C++ support does not force the programmer to use a specific object model, leaving him a high level of flexibility.

9.2.4 Improvements to Distributed File System

DFS has been enhanced, both with respect tofunctionality and performance. The latter was achieved by several separate developments, such as an optimized token manager and the introduction of an RPC mode that can transfer bulks of single data records in a single transfer. The DFS backup utilities now support tape stackers and jukeboxes.

- Optimized Token Manager

  A newly designed token manager in OSF DCE 1.2.1 reduces the memory need and improves token-related operations up to six times faster.

- Vnode/VM Management

  These improvements in OSF DCE 1.2.1 allow the Local File System (LFS) to perform significantly better as the system is subjected to higher levels of stress. The token manager will decrease the memory requirements and improve the performance and reliability of DFS.

- DFS Server Preferences

  DFS server selection by the clients was done in earlier OSF DCE releases with an algorithm that only took the IP address into account. With OSF DCE 1.2.1, clients have a more intelligent way to find best performing DFS servers, provided they have a selection of replicated servers to choose from. In addition, administrators can set preferences on a per-fileset bases to enforce optimal performance.

Because of its importance, IBM has implemented this server preferences mechanism already in AIX DCE Version 2.1 and DSS for OS/2 Warp before its availability in OSF DCE 1.2.1.

- Bulk Status RPC

  Directory operations, like getting file listings from a directory, are quite common and often needed during normal operation. OSF DCE 1.2.1 incorporates enhancements to RPC to transfer bulks of directory entries instead of one at a time as in earlier releases. Bulk status RPC can transfer up to 32 entries at a time and can improve overall directory operation performance by up to 50 percent.

- Improved Replication Performance

  The current algorithm in a DFS Replication Server works serially, which might introduce a bottleneck when heavily used. The enhancement in OSF DCE 1.2.1 is to handle requests in parallel, which improves performance and reliability in environments with many replicated DFS filesets.

- Enhanced Backup Utilities

  For automated backup of large amounts of data, the DFS backup utilities have been enhanced in OSF DCE 1.2.1 to support tape stackers and jukeboxes. DFS backup utilities can call external callout routines to perform tape operations needed to operate such tape-handling devices.

- DFS Backup Performance Improvements

  Several enhancements to the DFS backup procedures will improve its overall performance. This will be implemented in OSF DCE 1.2.2.

- Multi-Home Support

  The current implementation required a DFS File Server to be reachable through all its network interfaces. OSF DCE 1.2.2 will introduce methods to disable DFS from certain interfaces.

- Support for 64-Bit Filesystems

  DFS in OSF DCE 1.2.2 will support 64-bit filesystems while maintaining interoperability to existing 32-bit filesystems.

- Use of Protected RPC

  DFS in OSF DCE versions prior to 1.2 uses various protection levels for various kinds of traffic. Even unauthenticated RPC was used for certain traffic between repserver processes. DFS in OSF DCE 1.2.2 will eliminate all unauthenticated RPC and will allow the administrator to customize the protection levels for most communication within DFS, including different settings for inter-cell and intra-cell communications.

9.2.5 Other Enhancements

OSF DCE 1.2 incorporates a series of other enhancements:

Kerberos V5 Support and Kerberized Commands: Earlier versions of DCE already used MIT Kerberos Version 5 internally. OSF DCE 1.2.2 allows interoperability with other Kerberos V5 applications running on either DCE or non-DCE platforms to access the DCE Security Server as a Kerberos server. This is achieved by supporting IETF-RFC 1510.

In addition to this, two most often used commands for remote administration on UNIX platforms, `rlogin` and `rsh`, will be provided with OSF DCE 1.2.2 as DCE secured commands.

Public Key Support: OSF DCE 1.2.2 will add support for public key authentication methods while maintaining transparency for users not using public key methods. Public key technology frees the Security Server from the necessity to store (encrypted) passwords, thereby removing the thread from being compromised.

User-to-User Authentication: Prior to DCE 1.2, an application server program was required to have access to a local key store file on the machine on which it was running. This imposed some lack of flexibility and limited the use of server programs on diskless machines.

The proposed solution in OSF DCE 1.2.2 allows a server to run with only a login context, just as a client has when started after a user login to DCE.

Global Groups: In OSF DCE 1.2.2, users from foreign cells can be added to groups. This will ease administration in intercell installations.

Single-Threaded RPC: This simplification in OSF DCE 1.2.2 will allow easier portability of legacy applications to a DCE environment. Building a thread-safe environment in an existing, non-DCE application was sometimes very complicated or even impossible if those application used third-party tools. Single-Threaded RPC does not require thread-aware programming.

Scalability Improvements for the Security Server: In very large environments, say with 10,000+ users, the Security Server could have been a performance bottleneck. Some general improvements in OSF DCE 1.2.2 help to avoid such problems. Checkpointing will be configurable.

Documentation: OSF DCE 1.2.2 will ship documentation in industry-standard SGML (Standard Generalized Markup Language) format.

9.3 DCE Technology for the Internet and Intranet

Basically, the Internet is a TCP/IP-based network that provides the transport infrastructure so clients can communicate with their servers. The communication between clients and servers can be as simple as file sharing and terminal emulation or as complex as advanced client/server applications like Web browsers and Web servers. The Internet is a public network.

Intranet means the application of Internet technologies (such as Web browsers and servers) within an organization′s internal computer network (rather than on the worldwide public Internet). In other words, the intranet lets a company's employees access internal company information using the same interface they already use to access the global Internet.

Since the Internet/intranet is a TCP/IP based network and since DCE uses TCP/IP for communication (DCE can also use other communication protocols, but most often it is TCP/IP being used as communication protocol for DCE), the Internet/intranet technology supports DCE and vice versa.

Firewalls are used to separate the internal intranet from the public Internet. They generally support the use of DCE services across the intranet/Internet boundary. This means that DCE services can be made available for partners, customers, employees, and others who are located on the Internet and who are going to perform transactions into the servers located on an intranet. The DCE Security Service can even be used to encrypt the communication between the clients on the Internet and the servers on the intranet.

The Web is, and will become, even more a collection of distributed applications. Whereas Web browsers in their early days were something comparable to simple file browsers, they now are capable—together with sophisticated Web servers and integrated, server-based and client-based computing capabilities—of doing much more than just simple document browsing. On their intranets, the private internal networks of many institutions, companies have successfully begun to integrate their business applications into their Web. In fact, state-of-the-art Web servers and browsers incorporate many features required for most business applications:

- A graphical, easy-to-use user interface
- Many built-in helpers for special visibility enhancements, such as tables and frames
- Capability of drawing graphics with many special effects, like clickable images or animated pictures
- Gathering user input in various ways, such as plain text, radio buttons, selection lists, and so forth
- Processing based on user input
- Through links, easy integration of other services

IBM employees may find good examples of exploiting Web technology on an intranet for business support at `http://w3.austin.ibm.com/`. This IBM-internal home page of the Austin development site contains not only browsable information but also contains business request forms and the application logic behind them, as well as help desk accessibility and many other online forms.

Due to the nature of the Web, which basically makes any resource available to anybody, new security schemes must be applied on top of the network layers. Applications should not be required to deal with security management. They should, however, be able to "ask" a common security service whether a certain operation for a certain user is permitted or not.

This is, once more, a situation where DCE fits in perfectly. Products from IBM and Transarc for this purpose, bringing DCE Security to the Web, are:

- DFS Web from IBM (see also 6.7, "IBM DFS Web" on page 98)

  This solution incorporates DCE DFS Security on the server side, using a standard Web server.

- DE-Light Web Client from Transarc Corp.

  A client-only implementation using standard Web browsers and Java applets.

Even without specific products, DCE can be used to introduce strong security and access control features to the Web. An example is explained in "Simple DCE DFS Security for the Web" on page 99.

OSF has also developed code called *DCE Web* which is available to vendors that incorporates DCE security in an Internet and intranet Web environment.

9.4 More DCE Integration in IBM Products

DCE is a middleware product that acts as a distributed-application-enabling software. DCE on its own does not provide much valuable functionality for distributed systems, but it is a very powerful platform for applications that exploit its functions and services. DFS, the distributed file system closely related to DCE, is just one example of a product that uses DCE services underneath.

Customers can expect an increasing number of products and solutions from IBM that integrate DCE or can make use of its services. Chapter 6, "IBM′s Software Products Will Exploit DCE" on page 91 lists current products that already exploit DCE and may even experience further improvements in using DCE in their future releases and versions.

According to 1.4, "The IBM Open Blueprint" on page 5, DCE is the strategic platform for distributed applications from IBM. This is not only true for IBM, but also for many other major information technology providers, as can be seen in Chapter 4, "DCE Implementations on Non-IBM Platforms" on page 81. Customers can therefore expect more DCE integration in current and upcoming solutions and products, replacing proprietary, non-standard security implementations and directory services.

IBM will continue to prove its commitment to DCE technology by:

- Providing new, enhanced versions and releases of currently available products implementing DCE core functionality.
- Integration of strong DCE core functionality in existing, distributed applications.
- Providing professional services to its customers for the whole range, from initial consultancy to installation and implementation services through local and international.

9.4.1 Products Incorporating DCE Core Technology

IBM has been a founding member of the Open Software Foundation (OSF) and has since been among the first vendors incorporating OSF technology into its products. IBM technology has also been selected by the OSF for various functions that have become part of OSF's products.

IBM has particularly been participating in the development of OSF DCE 1.2 and customers can expect products from IBM that incorporate this new release of OSF. IBM selects those enhancements that customers require and will provide its own enhancements as has been the case in the past. On AIX, for example, IBM has added powerful administration tools for DCE on top of the standard administration commands of OSF DCE. And in the current version of DCE for AIX Version 4.X, IBM added an integrated login service for DCE and the operating system: Users do not need to be administrated on the AIX systems when they are defined in the DCE Registry.

Other examples of enhancements to the base OSF DCE functionality are the integration of the operating system specific, non-UNIX-like file systems of, for example, the OS/390, Windows NT, or OS/2 Warp platforms into DFS, either as clients or even as file servers.

Customers can expect enhancements, new versions and releases of DCE core technology on all of its platforms. The recent announcement of DCE for Windows

NT is another indication of IBM's commitment to DCE and customer requirements. DCE for Windows 95 will follow and IBM has stated that it will make available new versions of these products based on OSF DCE 1.1.

DCE for OS/400 and DSS for AIX are other platform products where customers can see IBM's commitment to DCE in the near future.

9.4.2 DCE Integration in Distributed Application-Enablers

IBM is currently evaluating the integration of DCE technology in other products. The most up-to-date information can be found on the IBM DCE Web pages, see A.2.1, "The IBM DCE Web Page" on page 149. The following application-enabling middleware and system support products are primary candidates for such enhancements:

- IBM DataBase 2

  The current support for DCE in IBM's DB2 products could be enhanced across all supported platforms to fully use DCE Security for authentication, secure RPC for data privacy and CDS for directory service. DB2 for AIX currently supports DCE Directory Services for database location lookup and DB2 for MVS/ESA has implemented new Distributed Relational Database Architecture (DRDA) functions that allow the use of DCE tickets for authenticating database clients. See also 6.3, "IBM DATABASE 2" on page 94.

- IBM MQSeries

  Similar to DB2 (see above), MQSeries could make full use of DCE by using DCE Security Services for authentication, secure RPC for data privacy and DCE Directory Services for services lookup functions. This would simplify and centralize its administration. See also 6.4, "IBM MQSeries" on page 95.

- IBM Single Sign-On

  The IBM Single Sign-On (SSO) product, already informally announced through press releases, will enable users to log on and to authenticate to different applications by entering their user ID and password on a single dialog window. Based on DCE Security Services and secure DCE RPC, SSO will then subsequently log the users in to the requested systems and applications. See also 6.6, "IBM Single Sign-On" on page 97 for more details.

- IBM DFS Web

  The upcoming IBM DFS Web product, informally announced through press releases, will incorporated DCE Security and DFS Access Control to Web-based applications. DFS Web will not require special Web browsers since it is a server-side implementation. Administration and auditing tools even allow easy maintenance and control over secure, distributed Web applications. More details can be found in 6.7, "IBM DFS Web" on page 98.

- IBM Parallel System Support Program (PSSP) for the SP2

  The IBM Parallel System Support Program for the IBM Scalable POWERparallel System (SP2) uses an MIT Kerberos implementation as its internal security system. A future release of the PSSP could make use of the DCE security system instead of using its own mechanisms.

- Lightweight Directory Access Protocol (LDAP) Support

IBM will add support for the Lightweight Directory Access Protocol to its main platform products, as stated in recent press releases. This will allow servers and clients to be more efficient, easier to configure and manage, and less resource consuming.

9.4.3 IBM Professional Services Organizations

IBM's Professional Services organizations and consulting groups are well educated and trained and are further being expanded for supporting customers in the broad area of distributed computing. This includes consulting for the design and implementation of distributed environments, as well as participating in projects for skills transfers to customers.

Professional Services can be requested though local IBM representatives. See also A.10, "Consultants and Services" on page 154.

Appendix A. Other Sources of Information

This appendix describes where and how to find out more about DCE. A lot of information is produced by The Open Group (formerly the Software Foundation and X/Open) and the vendor companies involved in DCE, including IBM and Transarc Corp. Most of it is available in electronic form on the Internet.

IBM employees may find more pointers to similar information, such as marketing material, many customer references, and addresses of consultant groups in IBM internal references. It is then up to the IBM representative to obtain the necessary permissions to pass some of this information on to customers. See A.7, "IBM Internal Documentation" on page 152 for more information.

Note: The Internet and its services are very dynamic; repositories may be closed or renamed, or a certain server may become temporarily unavailable. So, if you cannot find an object referenced in this redbook, try again later or look for another location that carries the same object.

A.1 Finding Information on the Internet

To find information on the Internet, you must have access to it. There are several ways to access the Internet, and your site may provide some or all of them. These are:

- Electronic mail or e-mail.
- News, a worldwide forum.
- Anonymous FTP service, allowing the user to retrieve files, programs, samples, articles, and so on by using the TCP/IP File Transfer Protocol and the *anonymous* login name.
- Gopher, a high-level mechanism to find information regardless of its format.
- WAIS, an index-based navigation tool in ASCII files.
- World Wide Web or WWW, a distributed hypertex-based networkt allowing access to many documents and protocols, including most of the ones mentioned above and support for images and sound. It has become very popular because of its power and sociability. The Web is sometimes confused with one of the most common programs used to access it, such as Mosaic.

If you know the name of an object that must be out on the Net, but you don′t know where to find it, try a keyword search on one of the search engines described in A.1.3, "Keyword Search Within the WWW" on page 149 below.

Another way to find interesting information is to start somewhere and follow the hyperlinks defined there.

A.1.1 Uniform Resource Locators (URLs)

To efficiently describe the access method and the location of a document, the concept of a Uniform Resource Locator (URL) has been created. Think of it as a networked extension of the standard filename concept: Not only can you point to a file in a directory, but that file and that directory can exist on any machine on the network, can be served via any of several different methods, and might not even be something as simple as a file. URLs can also point to queries, documents stored deep within databases, or to other objects.

For instance, suppose there is a document called foobar.txt; it sits on an anonymous ftp server called ftp.yoyodyne.com in directory /pub/files. The URL for this file is then `ftp://ftp.yoyodyne.com/pub/files/foobar.txt`. To get this document via the WWW, you enter the following URL as a location address in your favorite Web browser:

```
ftp://ftp.yoyodyne.com/pub/files/foobar.txt
```

You can obtain the same document by loging in to the anonymous ftp server and getting the file:

```
$ ftp ftp.yoyodyne.com
Connected to ftp.yoyodyne.com
Name (ftp:myname): anonymous
331 Guest login ok, send your complete e-mail address as password.
Password: myname@cougar.austin.ibm.com
230 Guest login ok, access restrictions apply
ftp> get pub/files/foobar.txt
```

Important other URLs are:

- Gopher URLs. To visit a particular gopher server (say, the gopher server on gopher.yoyodyne.com), use this URL: `gopher://gopher.yoyodyne.com/`
- News URLs. To point to a Usenet news group (say, rec.gardening), the URL is simply: `news:rec.gardening`
- HTTP URLs. They are used by the WWW. A file called foobar.html on HTTP server www.yoyodyne.com in directory /pub/files corresponds to this URL: `http://www.yoyodyne.com/pub/files/foobar.html`

Later in this appendix, we use URLs to point to information. If you experience problems accessing some services, you should read one of the numerous books, articles, magazines, and online help screens available, or refer to your administrator.

For some interesting URLs, such as IBM's DCE home page, see A.2, "DCE Information on the Internet" on page 149.

A.1.2 No Access to the Internet?

It is sometimes possible to access information by other ways, such as by mail, fax, or phone. We will note them when they are available, but they are rare.

However, because DCE is an open product, conceived and sold by several companies, there is no centralized source of information. So, you should really consider finding an access to the Internet. Today, individual low-cost accesses are becoming common. You should, for instance, look at the IBM solution based on OS/2 Warp.

A.1.3 Keyword Search Within the WWW

You can also use one of the WWW directory servers that allow you to search through the whole WWW by keyword. Some of these search engines are:

- http://www.yahoo.com/
- http://webcrawler.com/
- http://www.excite.com/
- http://www.isleuth.com/
- http://altavista.digital.com/
- http://www.einet.net/
- http://www.lycos.com/

If you don′t find something particular on one search engine, try another one. Most of these search engines and directories provide links to other such services on the Web.

A.2 DCE Information on the Internet

There are many sources of information about DCE on the World Wide Web. Because of the hypertext structure of the WWW, they often point to each other. Some of the most useful references are documented below.

A.2.1 The IBM DCE Web Page

IBM's DCE Web pages can be accessed through the URL http://www.raleigh.ibm.com/dce/dcehome.html. This IBM DCE home page contains a lot of information on current and upcoming products. On these pages, IBM even sometimes provides information about products that have not formally been announced. These pages contain the most recent product information and can therefore be regarded as a supplement to this redbook. The IBM DCE Web pages also serve as a good source for interesting general documents on DCE, white papers and various reports, including customer or vendor quotes and success stories or DCE performance reports.

Finally, it gives you an overview on IBM service offerings, such as consulting, for DCE-related customer projects.

Another IBM Web page worth looking at with its many linked subpages is located at URL http://www.software.ibm.com/openblue/. It describes the IBM Open Blueprint and contains its latest architectural updates.

There are, of course, many other IBM Web pages that we do not explicitly mention here. A good starting point to explore IBM's Web pages is the home page at http://www.ibm.com/.

A.2.2 The Open Group WWW Server

The Open Group's home page is located at URL http://www.opengroup.org/. It was created after the merger of Open Software Foundation and X/Open. This page contains links to all kind of information. Some special services they provide through their Web pages are:

The Open Software Registry
This is a database of open software products available from various vendors. Search tools help to find certain software. See also A.3, “The Open Software Registry” on page 151.

The Software Mall
This lets you download and evaluate various software and documentation developed by The Open Group (or formerly the Open Software Foundation) and others.

Some services or information require you to register; others may even be available only to members.

At the time this book was written, the merger of the Open Software Foundation (OSF) and X/Open has not yet been completed on their WWW servers. Due to this, not all information pertinent to DCE is available on their new home page, `http://www.opengroup.org/`, but may still be found under `http://www.osf.org/`. Over time, it can be assumed that the latter URL will disappear.

A.2.3 The OSF WWW Server

The Open Software Foundation still maintains a WWW server with information about all the OSF products, including DCE. The home page is `http://www.osf.org`, but `http://www.osf.org/dce/index.html` takes you directly to the DCE index. Highlights of the OSF Web server include a complete list of DCE products (`http://www.osf.org/comm/lit/dce-prod-cat/`) and a hypertext version of the Frequently Asked Questions list (`http://www.osf.org/dce/qna/index.html`). The URL `http://www.osf.org/comm` also contains useful information about DCE. Most of the OSF literature found on this server is available for reprint.

A.2.4 The News Groups

A news group is dedicated to DCE. It is `news:comp.soft-sys.dce`. It is very active on technical issues, but sometimes addresses more general issues. Other interesting news groups are:

- `news:comp.client-server`, which has interesting discussions on many aspects of client/server computing, including DCE, CORBA, languages of choice, and so on
- `news:comp.unix.osf.misc`, which used to be active on DCE before the creation of `news:comp.soft-sys.dce` and still has some discussions on it

A.2.5 CITI Technical Reports

The Center for Information Technology Integration (CITI) at the University of Michigan has several pieces of DCE information on their servers. The big attraction here is the set of CITI Technical Reports related to DCE. You can get to the reports from the CITI home page, using `http://www.citi.umich.edu/`, or directly by using the URL `http://www.citi.umich.edu/techreports/`.

A.2.6 Miscellaneous

The following is a list of miscellaneous sources of information with a short description of their purpose:

- The OSF address `direct@osf.org` can be used to ask questions of the form ″Where can I get a XXXX for DCE?″ You can use the phone number (617) 621-7300 (USA) for the same purpose, or write to:

OSF Direct
11 Cambridge Center
Cambridge, MA 02142-1405 USA

- http://www.fokus.gmd.de/ovma/dooos/dce.html is an index page pointing to many of the DCE-related documents on the Web.
- Transarc Corporation has, like many other companies, interesting information and documents available through their Web pages at http://www.transarc.com/.
- Project Pilgrim at the University of Massachusetts has a DCE home page at http://info.pilgrim.umass.edu/. Their server provides a directory of contributed software, currently consisting of performance measurement utilities from Pilgrim, book examples from O′Reilly & Associates, as well as a complete set of RFCs, searchable by http.
- Many companies working in the DCE field have at least a mail address and sometimes an ftp server or a WWW page. The OSF DCE Product Catalog gives these addresses.
- Most book publishers have searchable book indexes on the Web. For DCE-related books, O' Reilly & Associates (http://www.ora.com/www/index.html) and Prentice Hall (http://www.prenhall.com/) are good addresses.

A.3 The Open Software Registry

The Open Software Registry is a database with all kinds of software, maintained by The Open Group. Vendors are invited to provide their product information to be added to this registry. Search tools are available on their Web page to selectively search after specific information, such as vendor name or the type (sector) of software. The registry houses information not only on DCE products, but DCE is certainly one of the selection criteria provided.

The Open Software Registry can be accessed through the URL http://www.opensoftware.com/. Users, as well as vendors, are required to be registered in order to get access to the database.

A.4 The OSF DCE Product Catalog

The OSF maintains a list of products related to DCE. This includes DCE implementations, DCE applications and tools and DCE services. It is a good tool for finding short descriptions of available products and contacts and therefore gives a view of the current DCE world.

The OSF DCE Product Catalog can be accessed from the OSF WWW server at the URL http://www.osf.org/dce/index.html, or directly by its name http://www.osf.org/comm/lit/dce-prod-cat.

The OSF DCE Product Catalog might be integrated into The Open Group's Open Software Repository in the future.

A.5 Frequently Asked Questions

The Frequently Asked Questions (or FAQ) is a list of questions and answers covering many aspects of DCE. Due to their informal nature, they contain a lot of practical and useful information, put together by experienced authors. It is a valuable collection of information, giving pointers to products, Internet servers, books, articles, and technical advice.

There are several places where FAQs can be found. An interesting FAQ about DCE is available through URL http://www.osf.org/dce/faq-mauney.html. It contains many links to other places of information or even addresses for document and program download.

The FAQ is probably the best starting point to find vendor-independent information on DCE.

A.6 International DCE Users Groups

The formation of the International DCE Users Group (IDCEUG) was announced on August 31, 1994. The IDCEUG is open to any computer professional and is not affiliated with The Open Group, OSF, or any other organization. Its purpose is to provide a forum for exchange of information among those who are applying DCE technology and ultimately to influence vendor products supporting the technology.

There are DCE Users Groups in North America and Europe. A list of contacts for the various DCE Users Groups by location can be found on the Web under http://www.osf.org/dce/user-groups/index.html. It can be assumed that this list will be moved to the Web pages of The Open Group at a certain time.

A.7 IBM Internal Documentation

In addition to all the information available to the public, IBM employees can access additional resources available on the worldwide internal networks and databases.

A.7.1 VM Tools Disk

If you are working for IBM, you can access marketing or technical documentation from your VM account. These packages are available from the MKTTOOLS tools disk. To obtain a list of packages, type the following command from the VM command line:

```
TOOLS SENDTO USDIST MKTTOOLS MKTTOOLS LIST * *
```

Then, to obtain a certain package, type:

```
TOOLS SENDTO USDIST MKTTOOLS MKTTOOLS GET package-name PACKAGE
```

A better way to access the MKTTOOLS catalog is to use the interactive access method, if it is provided on your VM system:

```
TOOLCAT MKTTOOLS
```

The packages that can be found on this tools disk are presentations, summary documents, and product descriptions. We are not going to list single packages since this is a dynamic disk; new packages are added, old packages are removed in an ongoing process.

A.7.2 Internal Web Pages and News Groups

The IBM Internal DCE Web home page is located at URL http://w3.austin.ibm.com/dce/. It contains not only links and information about IBM's internal huge DCE and DFS installations and administration, but it also carries links to various kind of documents.

Some of the pages behind links from these pages may be protected and not available to any IBMer.

For those IBMers who have access to the IBM news groups, there are several groups at news:ibm.awd.dce.* carrying up-to-date and interesting information about DCE. They can certainly also be used for asking questions.

A.8 Customer References

The OSF has a list of customer references, no matter what vendor they use. They are called *End-User Profiles* and are available on the WWW page at URL http://www.osf.org/comm/lit/dce-eup.

The *Case Studies and Histories* part of this server also contains interesting customer use of DCE (see URL http://www/osf/org/lit/lit-cases.html).

Also, on the same WWW server, *Production Stories* can be found under URL http://www.osf.org/comm/lit/cwdcemag/. They explain, in more details, specific customer environments and the solution achieved with DCE technology.

IBM maintains a list of customer references on its DCE Web page at URL http://www.raleigh.ibm.com/dce/dcehome.

A.9 The Developer Connection CD-ROM

This CD-ROM is made by IBM and contains a lot of tools, documentation, and evaluation copies of products primarily related to OS/2 LAN systems. Some of them are related to DCE on OS/2 and AIX platforms and are very interesting. This includes:

- The MakeDCE tool and its documentation
- Other DCE tools, including ezDCE and PDDA6000
- A lot of documentation about DCE, including redbooks
- The valuable white papers described in A.7, "IBM Internal Documentation" on page 152

Subscription to the Developer Connection is available at the following phone numbers:

- United States: 1-800-6DEVCON (1-800-633-8266)
- Canada: 1-800-561-5263
- Brazil: 0800-111205
- Mexico: +91-800-00639 (627-2444 within Mexico City)

- Asia/Pacific: +61-2-354-7684 (61 is the country code for Australia)
- Europe, Middle East, Africa and Latin America (EMEA):

  Operators sitting in the publication center in Denmark and speaking the following languages are available: (45 is the country code for Denmark)

  - Dutch: +45-48101400
  - French: +45-48101200
  - Italian: +45-48101600
  - English: +45-48101500
  - German: +45-48101000
  - Spanish: +45-48101100

A.10 Consultants and Services

There are several consultant and services groups available that provide professional DCE support across country borders or even worldwide:

- The *Integrated Systems Solutions Corporation* (ISSC)

  The ISSC has the mission of delivering high-quality expertise and consulting services. The ISSC Open Systems Consulting Group provides DCE consulting across all vendor platforms. The types of services they offer range from design evaluations to complete turn-key solutions. More information can be found on the Web at URL `http://www.rs6000.ibm.com/issc_oscs/`.

- DCE Cross-Platform Expert Line

  A cross-platform DCE expert line is available for customers or IBM marketing. It can be reached by e-mail by the address `odcm@austin.ibm.com`. It can be contacted for requesting expertise on DCE issues.

Customers should contact their local IBM representative if they need help from the above-mentioned services. Also, the local IBM representatives may be able to help with other, local support.

Appendix B. Special Notices

This publication is intended to help customers, system engineers, and marketing representatives understand IBM′s current and future DCE offering on all the platforms, what tools for administration and development are available and where to find more information. The information in this publication is not intended as the specification of any programming interfaces that are provided by IBM′s OS/390 DCE, IBM′s OpenEdition DCE for VM/ESA, IBM′s DCE Base Serivces/400, IBM′s DCE for AIX Version 2.1 Product Family, or IBM′s DSS for OS/2 Warp and DCE for Windows products. See the PUBLICATIONS section of the IBM Programming Announcement for IBM′s OS/390 DCE, IBM′s VM/ESA OpenEdition DCE, IBM′s DCE Base Serivces/400, IBM′s DCE for AIX Version 2.1 Product Family, or IBM′s DSS for OS/2 Warp and DCE for Windows products for more information about what publications are considered to be product documentation.

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 vendors or is made available to the public by the vendors, 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:

AD/Cycle
AIX
AIX/6000
AnyNet
APPN
AS/400
C/370
C/400
CICS
CICS/ESA
CICS/6000
COBOL/370
COBOL/400
DATABASE 2
DB2
DB2/2
DFSMS/MVS
DFSMS/VM
Distributed Relational Database Architecture
DRDA
HACMP/6000
IBM
IMS/ESA
InfoExplorer
LANDP
Language Environment
MQSeries
MVS/ESA
NetView
OpenEdition
OS/2
OS/400
PS/2
RACF
RISC System/6000
RS/6000
SOMobjects
VisualAge
VM/ESA
VTAM

The following terms are trademarks of other companies:

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.

Java and HotJava are trademarks of Sun Microsystems Inc.

AT&T, AT&T GIS	AT&T
ANSAware	APM, Ltd.
Cell Manager	HaL Computer Systems, Inc.
CodeCenter, ObjectCenter	CenterLine Software, Inc.
Connection/DCE	Open Horizon, Inc.
CORBA	Object Management Group, Inc.
DG/UX	Data General Corporation
DCE, Motif, Open Software Foundation, OSF/1, OSF	The Open Software Foundation
DEC, VMS, DIGITAL, VT100	Digital Equipment Corporation
Ellery Open Systems	Ellery Systems, Inc.
Episode, Encina, Transarc	Transarc Corporation
HP, HP/UX	Hewlett-Packard Company
Insure++	ParaSoft Corporation
INFORMIX	Informix Software, Inc.
Ingres	Computer Associates International, Inc.
Macintosh	Apple Computer, Inc
Micro Focus	Micro Focus Limited
NetWare, Novell, UnixWare, IPX	Novell, Inc.
Network File System, NFS, NIS, ONC, Solaris, Sun, SunOS	Sun Microsystems, Inc.
ORACLE	Oracle Corporation
OEC, Encompass	Open Environment Corporation
POSIX	Institute of Electrical and Electronic Engineers
PowerBuilder	PowerSoft Corporation
Purify, Quantify	Pure Soft, Inc.
santix, DCEmgmt	santix software GmbH
SCO	The Santa Cruz Operation, Inc.
Siemens, Siemens-Nixdorf, Sinix	Siemens Aktiengesellschaft
SNAP	Template Software, Inc.
Sybase	Sybase, Inc.
Tandem	Tandem Computers, Inc.
Tivoli, TME 10	Tivoli Systems Corp.
X/Open	X/Open Company Limited
VisualRPC, CROSSLOGIC	CROSSLOGIC Corporation
Visual-DCE	Gradient Technologies, Inc.
Windows, Windows NT, Microsoft Windows, Visual Basic, Visual C++	Microsoft Corporation

Other trademarks are trademarks of their respective companies.

Appendix C. Related Publications

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

C.1 International Technical Support Organization Publications

For information on ordering these ITSO publications, see “How To Get ITSO Redbooks” on page 165.

- *Administering IBM DCE and DFS Version 2.1 for AIX (and OS/2 Clients)*, SG24-4714
- *Developing DCE Applications for AIX, OS/2 and Windows*, GG24-4090
- *The Distributed File System (DFS) for AIX/6000*, GG24-4255
- *Understanding OSF DCE 1.1 for AIX and OS/2*, SG24-4616
- *Using and Administering AIX DCE 1.3*, GG24-4348
- *MVS/ESA OpenEdition DCE: Application Support Servers CICS and IMS*, GG24-4482
- *MVS/ESA OpenEdition DCE: Application Development Cookbook*, GG24-4481
- *MVS/ESA OpenEdition DCE: Installation and Configuration Experiences*, GG24-4480
- *MVS/ESA OpenEdition DCE Presentation Guide Volume I*, Available on the SK2T-2177-09 CD-ROM
- *OpenEdition MVS for MVS/ESA 5.1: Presentation Guide*, Available on the SK2T-2177-09 CD-ROM
- *MVS/ESA OpenEdition DCE: RACF and DCE Security*, GG24-2526
- *OpenEdition for VM/ESA Implementation and Administration Guide*, SG24-4747
- *VM/ESA OpenEdition DCE Introduction and Implementation Notebook*, SG24-4554
- *VM/ESA Version 2 Overview and Technical Guide*, GG24-4418
- *Client/Server Computing with VM/ESA as Part of the Enterprise*, GG24-3950
- *Developing (Real) DCE Applications for OS/400*, SG24-2572
- *Elements of Security: AIX 4.1*, GG24-4433
- *Inside the Directory and Security Server for OS/2 Warp, SG24-4785 (in press)*
- *LANDP/6000 Concepts and Guidelines*, GG24-4057
- *DCE Cell Design Considerations*, SG24-4746

C.2 Redbooks on CD-ROMs

Redbooks are also available on CD-ROMs. **Order a subscription** and receive updates 2-4 times a year at significant savings.

CD-ROM Title	**Subscription Number**	**Collection Kit Number**
System/390 Redbooks Collection	SBOF-7201	SK2T-2177
Networking and Systems Management Redbooks Collection	SBOF-7370	SK2T-6022
Transaction Processing and Data Management Redbook	SBOF-7240	SK2T-8038
AS/400 Redbooks Collection	SBOF-7270	SK2T-2849
RISC System/6000 Redbooks Collection (HTML, BkMgr)	SBOF-7230	SK2T-8040
RISC System/6000 Redbooks Collection (PostScript)	SBOF-7205	SK2T-8041
Application Development Redbooks Collection	SBOF-7290	SK2T-8037
Personal Systems Redbooks Collection	SBOF-7250	SK2T-8042

C.3 Other Publications

These publications are also relevant as further information sources:

General DCE Books

The following are general IBM documentation about DCE:

- *Networking Blueprint, Executive Overview*, GC31-7057
- *Open Blueprint Technical Overview*, GC23-3808
- *Distributed Computing Environment: Understanding DCE Concepts*, GC09-1478

The following documentation from OSF (The Open Group) can be ordered either through Prentice Hall or directly through The Open Group:

- *OSF DCE Version 1.0.3 Documentation Set - 14 Volumes*
- *OSF DCE Version 1.1 Documentation Set - 13 Volumes*
- *Introduction to OSF DCE Release 1.1*
- *OSF DCE Administration Guide -Release 1.1 Volume 1: Introduction*
- *OSF DCE Administration Guide Volume 2 Core Components Release 1.1*
- *OSF DCE Application Development Reference - Volume 1 Release 1.1*
- *OSF DCE Application Development Guide Release 1.1 Volume 1*
- *OSF DCE Application Development Guide - Core Compts Release 1.1*
- *OSF DCE Application Development Guide - Dir Services Release 1.1*
- *OSF DCE GDS Administration Guide - Release 1.1 Extended Services*
- *OSF DCE Application Development Reference - Volume 1 Release 1.0*

The following books are published by O'Reilly & Associates:

- *Understanding DCE*
- *Guide to Writing DCE Applications*
- *DCE Security Programming*
- *Pthreads Programming*

- *Distributing Applications Across DCE and Windows NT*

OS/390 Platform

- *IBM Online Library Omnibus Edition: OS/390 Collection*, SK2T-6700
- *OS/390 OpenEdition DCE: Introduction*, GC28-1581
- *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: Intro and Style*, SC28-1587
- *OS/390 OpenEdition DCE Application 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
- *OS/390 OpenEdition DCE: DFS Administration Guide and Reference*, SC28-1720
- *OS/390 OpenEdition DCE: DFS Command Reference*, SC28-1721
- *OS/390 OpenEdition DCE: DFS Configuring and Getting Started*, SC28-1722
- *OS/390 OpenEdition DCE: DFS Messages and Codes*, SC28-1724
- *Introducing OS/390 OpenEdition DCE DFS*, GC28-1723
- *OS/390 Security Server (RACF) Support for OpenEdition DCE*, GC28-1924
- *OS/390 Security Server (OpenEdition DCE Security Server) Overview*, GC28-1938

VM Platform

- *IBM Online Library Omnibus Edition: VM Collection*, SK2T-2067
- *IBM OpenEdition DCE for VM/ESA: Introducing the OpenEdition Distributed Computing Environment*, SC24-5735
- *IBM OpenEdition DCE for VM/ESA: Planning*, SC24-5737
- *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: Administration Reference*, SC24-5731
- *IBM OpenEdition DCE for VM/ESA: Administration Guide*, SC24-5730
- *IBM OpenEdition DCE for VM/ESA: Users Guide*, SC24-5738
- *IBM OpenEdition DCE for VM/ESA: Configuring and Getting Started*, SC24-5734

- *IBM OpenEdition DCE for VM/ESA: Messages and Code*, SC24-5736

OS/400 Platform

- *AS/400 Softcopy Library V3R2*, SK2T-2171
- *Introducing DCE/400*, GC09-1710
- *DCE: Planning*, SC09-1711
- *DCE: Application Development*, SC09-1713
- *DCE: Application Development Reference*, SC09-1714
- *DCE: Administration Reference*, SC09-1716
- *DCE: Administration Guide*, SC09-1715
- *DCE: Configuring and Getting Started*, SC09-1712

AIX Platform

AIX DCE 2.1 and DSS for AIX comes with printable online documentation. The only separately orderable documents are the following:

- *AIX DCE Directory and Security Services Up and Running*, SC23-1797
- *Introduction to DCE V2.1 for AIX,* SC23-2796
- *DCE V2.1 for AIX: Getting Started*, SC23-2797

The product CD-ROM contains softcopy versions of the following books:

- *Introduction to DCE*
- *Directory and Security Server for AIX Up and Running*
- *DCE for AIX NFS/DFS Authenticating Gateway*
- *DCE for AIX Application Development Guide -- Introduction and Style Guide*
- *DCE for AIX Application Development Guide -- Core Components*
- *DCE for AIX Application Development Guide -- Directory Services*
- *DCE for AIX Administration Command Reference*
- *DCE for AIX Administration Guide*
- *DCE for AIX Application Development Reference*
- *DCE for AIX DFS Administration Guide and Reference*

Documents for earlier versions of DCE for AIX:

- *DCE V1.3 for AIX User's Guide and Reference*, SC23-2729
- *DCE V1.3 for AIX Administration Guide -- Core Services*, SC23-2730
- *DCE V1.3 for AIX Administration Guide -- Extended Services*, SC23-2731
- *DCE V1.3 for AIX Administration Reference*, SC23-2732
- *DCE V1.3 for AIX Application Development Guide*, SC23-2733
- *DCE V1.3 for AIX Application Development Reference*, SC23-2734
- *DCE NFS to DFS Authenticating Gateway V1.3 for AIX*, SC23-2735
- *NetView for DCE and Encina Manager Guide V1.3*, SC23-2736
- *AIX HACMP for DCE and Encina Guide V1.3*, SC23-2737

- *AIX DCE Getting Started V1.3*, SC23-2477
- *AIX DCE and OS/2 DCE Message Reference*, SC23-2583

OS/2 and Windows Platform

DSS for OS/2 Warp comes with printable online documentation. The product CD-ROM includes:

- *DSS Up and Running*
- *DSS Administrator′s Reference*
- *DSS Command and Utilities*
- *DSS Client User′s Guide*
- *DSS Problem Determination*
- *DSS Programming Guide and Reference*
- *DSS Trademarks*
- *MPTS Configuration Guide*
- *MPTS Sockets Programming Reference*
- *DCE Getting Started*
- *DCE Administration Guide*
- *DCE Administration Commands Reference*
- *DCE Application Development Reference*
- *DCE Application Development Guide — Introduction and Style*
- *DCE Application Development Guide — Core*
- *DCE Application Development Guide — Directory Services*
- *DCE DFS Client Guide and Reference*
- *DCE Error Message Manual*
- *DCE Glossary*

Books for earlier versions of DCE for OS/2 and Windows are:

- *IBM DCE for OS/2: Guide to Planning, Installation and Configuration*, S96F-8502
- *IBM DCE for OS/2: Administrator's Guide*, S96F-8504
- *IBM DCE for OS/2: Administrator's Command Reference*, S96F-8505
- *IBM DCE for OS/2: Application Developer's Guide*, S96F-8506
- *IBM DCE for OS/2: Application Developer's Reference*, S96F-8507
- *IBM DCE for OS/2: Master Index*, S96F-8615
- *IBM DCE Client for Windows User's Guide*, S96F-8622
- *IBM DCE SDK for Windows Guide and Reference*, S96F-8623

DCE for Windows NT includes online documentation on the product media CD-ROM.

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`.

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 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`

- **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.

- **REDBOOKS 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) — send orders to:

	IBMMAIL	**Internet**
In United States:	usib6fpl at ibmmail	usib6fpl@ibmmail.com
In Canada:	caibmbkz at ibmmail	lmannix@vnet.ibm.com
Outside North America:	dkibmbsh at ibmmail	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 29570
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	1-403-267-4455
Outside North America	(+45) 48 14 2207 (long distance charge)

- **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

- **On the World Wide Web**

Redbooks Home Page	http://www.redbooks.ibm.com
IBM Direct Publications Catalog	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

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
ADSM	ADSTAR Data Storage Management
ANSI	American National Standards Institute
API	Application Programming Interface
ATM	asynchronous transfer mode
CDMF	Commercial Data Masking Facility
CDS	Cell Directory Service
CICS	Customer Information Control System
CMIP	common management interface protocol
CMVC	Configuration Management and Version Control
CORBA	common object request broker
COSE	Common Open Software Environment
CSP	Cross-System Product
CTS	Common Transport Semantics
DES	data encryption standard
DFSMS	
DCE	Distributed Computing Environment
DAP	directory access protocol
DDCS	distributed database connection service
DES	data encryption standard
DFS	Distributed File System
DME	Distributed Management Environment
DNS	domain name service
DRDA	distributed relational database architecture
DSM	Distributed Security Manager
DSOM	distributed system object model
DSS	Directory and Security Server
EPAC	extended privilege attribute certificate
ERA	extended registry attributes
FCS	fibre channel standard
FLDB	Fileset Location Database
GDA	Global Directory Agent
GDF	group definition file
GDS	Global Directory Service
GSS-API	Generic Security Service API
HACMP	High Availability Cluster Multi-Processing
HTTP	HyperText Transfer Protocol
IBM	International Business Machines Corporation
ICC	Initial Container Creation (ACL)
IDL	interface definition language
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IHMP	IBM NetView Hub Management Program
ILE	integrated language environment
IOC	Initial Object Creation (ACL)
IP	Internet Protocol
IPF	Information Presentation Facility (OS/2)
ISO	International Standardization Organization
ISSC	Integrated Systems Solutions Corporation
ITSO	International Technical Support Organization
LAN	local area network
LANDP	LAN Distributed Platform
LDAP	Lightweight Directory Access Protocol
LFS	Local File System
LSE	LAN Server Enterprise
MAN	metropolitan area network
MQI	message queuing interface
MPTN	Multiprotocol Transport Networking
MRPC	Microsoft RPC
NFS	Network File System
NIS	Network Information System
NTP	network time protocol
OLTP	on-line transaction processing
OMG	Object Management Group
ONC	Open Network Computing
OSF	Open Software Foundation
PAC	privilege attribute certificate
PTX	Performance Toolbox
RACF	Resource Control Access Facility (MVS)
RDBMS	relational database management system
RPC	remote procedure call
SAF	System Authorization Facility (MVS)
SCM	System Control Machine
SGML	Standard Generalized Mark-up Language
SLC	Secured Logon Coordinator (NetSP)
SNG	Secured Network Gateway (NetSP)
SNMP	simple network management protocol

SOM	system object model
SQL	structured query language
SSSO	(IBM) Secure Single Sign-On
TCL	Tool Control Language
TCP	transmission control protocol
TPI	time provider interface
TSO	Time Sharing Option (MVS)
UDMF	User Data Masking Facility
UDP	User Datagram Protocol
URL	Uniform Resource Locator
UUID	universal unique identifier
WAN	wide area network
XMP	X/Open management protocol

Index

Printed in U.S.A.

SG24-2543-01