

z/OS



# DCE Introduction



z/OS



# DCE Introduction

**Note**

Before using this information and the product it supports, be sure to read the general information under Appendix A, "Notices" on page 29.

**First Edition (March 2001)**

This edition, GC24-5911-00, applies to Version 1 Release 1 of z/OS DCE Base Services, z/OS DCE user Data Privacy (DES and CDMF), z/OS DCE User Data Privacy (CDMF) (program number 5694-A01), and to all subsequent releases and modifications until otherwise indicated in new editions.

Order publications through your IBM representative or the IBM branch office serving your locality. Publications are not stocked at the address given below.

IBM welcomes your comments. A form for reader's comments may be provided at the back of this publication, or you may address your comments to the following address:

International Business Machines Corporation  
Information Development, Dept. G60  
1701 North Street  
Endicott, NY 13760-5553  
United States of America

FAX (United States & Canada): 1+607+752-2327  
FAX (Other Countries):  
Your International Access Code +1+607+752-2327

IBMLink™ (United States customers only): GDLVME(PUBRCF)  
Internet e-mail: pubrcf@vnet.ibm.com  
World Wide Web: <http://www.ibm.com/servers/eserver/zseries/zos/>

If you would like a reply, be sure to include your name, address, telephone number, or FAX number.

Make sure to include the following in your comment or note:

- Title and order number of this book
- Page number or topic related to your comment

When you send information to IBM, you grant IBM a nonexclusive right to use the information in any way it believes appropriate without incurring any obligation to you.

© **Copyright International Business Machines Corporation 1994, 2001. All rights reserved.**

Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corp.

The following statements are provided by the Open Software Foundation.

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

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

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

Copyright © 1993, 1994 Open Software Foundation, Inc.

This documentation and the software to which it relates are derived in part from materials supplied by the following:

- © Copyright 1990, 1991 Digital Equipment Corporation
- © Copyright 1990, 1991 Hewlett-Packard Company
- © Copyright 1989, 1990, 1991 Transarc Corporation
- © Copyright 1990, 1991 Siemens Nixdorf Informationssysteme AG
- © Copyright 1990, 1991 International Business Machines Corporation
- © Copyright 1988, 1989 Massachusetts Institute of Technology
- © Copyright 1988, 1989 The Regents of the University of California

All Rights Reserved.

Printed in the U.S.A.

THIS DOCUMENT AND THE SOFTWARE DESCRIBED HEREIN ARE FURNISHED UNDER A LICENSE, AND MAY BE USED AND COPIED ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE INCLUSION OF THE ABOVE COPYRIGHT NOTICE. TITLE TO AND OWNERSHIP OF THE DOCUMENT AND SOFTWARE REMAIN WITH OSF OR ITS LICENSORS.

Open Software Foundation, OSF, the OSF logo, OSF/1, OSF/Motif, and Motif are trademarks of the Open Software Foundation, Inc.

UNIX is a registered trademark of The Open Group in the United States and other countries.

DEC, DIGITAL, and ULTRIX are registered trademarks of Digital Equipment Corporation.

DECstation 3100 and DECnet are trademarks of Digital Equipment Corporation.

HP, Hewlett-Packard, and LaserJet are trademarks of Hewlett-Packard Company.

Network Computing System and PasswdEtc are registered trademarks of Hewlett-Packard Company.

AFS and Transarc are registered trademarks of the Transarc Corporation.

Episode is a trademark of the Transarc Corporation.

Ethernet is a registered trademark of Xerox Corporation.

DIR-X is a trademark of Siemens Nixdorf Informationssysteme AG.

MX300i is a trademark of Siemens Nixdorf Informationssysteme AG.

NFS, Network File System, SunOS and Sun Microsystems are trademarks of Sun Microsystems, Inc.

X/Open is a trademark of The Open Group in the U.K. and other countries.

PostScript is a trademark of Adobe Systems Incorporated.

FOR U.S. GOVERNMENT CUSTOMERS REGARDING THIS DOCUMENTATION AND THE ASSOCIATED SOFTWARE

These notices shall be marked on any reproduction of this data, in whole or in part.

**NOTICE:** Notwithstanding any other lease or license that may pertain to, or accompany the delivery of, this computer software, the rights of the Government regarding its use, reproduction and disclosure are as set forth in Section 52.227-19 of the FARs Computer Software-Restricted Rights clause.

**RESTRICTED RIGHTS NOTICE:** Use, duplication, or disclosure by the Government is subject to the restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 52.227-7013.

**RESTRICTED RIGHTS LEGEND:** Use, duplication or disclosure by the Government is subject to restrictions as set forth in paragraph (b)(3)(B) of the rights in Technical Data and Computer Software clause in DAR 7-104.9(a). This computer software is submitted with "restricted rights." Use, duplication or disclosure is subject to the restrictions as set forth in NASA FAR SUP 18-52.227-79 (April 1985) "Commercial Computer Software-Restricted Rights (April 1985)." If the contract contains the Clause at 18-52.227-74 "Rights in Data General" then the "Alternate III" clause applies.

US Government Users Restricted Rights—Use, duplication or disclosure restricted by GSA ADP Schedule Contract.

Unpublished—All rights reserved under the Copyright Laws of the United States.

This notice shall be marked on any reproduction of this data, in whole or in part.



---

# Contents

<b>About This Book</b> . . . . .	ix
Where to Find More Information . . . . .	ix
Softcopy Publications . . . . .	ix
Internet Sources . . . . .	ix
Using LookAt to Look up Message Explanations . . . . .	ix
Accessing Licensed Books on the Web . . . . .	x
<b>Chapter 1. Distributed Computing</b> . . . . .	1
The Need . . . . .	1
The Solution . . . . .	2
The Benefits . . . . .	2
Open Systems . . . . .	3
The Open Software Foundation . . . . .	3
IBM and DCE . . . . .	4
<b>Chapter 2. DCE Features</b> . . . . .	7
The Client/Server Model . . . . .	7
Overview of DCE . . . . .	8
DCE Architecture . . . . .	9
Integrating the DCE Components . . . . .	11
<b>Chapter 3. z/OS DCE</b> . . . . .	13
Exploiting System Strengths . . . . .	13
Existing Cooperative Processing Applications . . . . .	14
Using DCE for Online Transaction Processing (OLTP) . . . . .	14
Accessing CICS and IMS . . . . .	15
z/OS DCE Architecture . . . . .	15
DCE Participation . . . . .	16
<b>Chapter 4. Application Support Servers</b> . . . . .	19
Application Support Transactional RPCs for IMS and CICS . . . . .	20
<b>Chapter 5. Working in a Distributed Computing Environment</b> . . . . .	21
Using DCE . . . . .	21
z/OS DCE Scenarios . . . . .	23
In a Manufacturing Environment . . . . .	23
In an Educational Environment . . . . .	25
Summary . . . . .	27
<b>Appendix A. Notices</b> . . . . .	29
Trademarks . . . . .	30
<b>Glossary</b> . . . . .	31
<b>Bibliography</b> . . . . .	37
z/OS DCE Publications . . . . .	37
z/OS SecureWay® Security Server Publications . . . . .	37
Tool Control Language Publication . . . . .	38
IBM C/C++ Language Publication . . . . .	38
z/OS DCE Application Support Publications . . . . .	38

Encina Publications . . . . .	39
<b>Index</b> . . . . .	<b>41</b>



---

## Figures

1.	A Typical Complex Computing Environment . . . . .	1
2.	Making the Connection with a Distributed Computing Environment . . . . .	2
3.	End User's View . . . . .	3
4.	The Interoperability Components . . . . .	5
5.	Client/Server Communication . . . . .	7
6.	Complex System Connections . . . . .	8
7.	The DCE Connection . . . . .	8
8.	The OSF DCE Architecture . . . . .	9
9.	DCE's Integrated Components . . . . .	12
10.	z/OS in a Distributed Computing Environment . . . . .	13
11.	z/OS DCE Client/Server Support . . . . .	14
12.	z/OS DCE Architecture . . . . .	16
13.	z/OS Cell Participation . . . . .	17
14.	Application Support server . . . . .	19
15.	Views of z/OS DCE . . . . .	22
16.	Leanside's Existing Environment . . . . .	24
17.	Leanside's New DCE Environment . . . . .	24
18.	The Music Information Network's DCE Environment . . . . .	27



---

## About This Book

This book introduces you to z/OS DCE. Whether you are a system manager, technical planner, z/OS system programmer, or application programmer, it will help you understand the Distributed Computing Environment (DCE).

To get acquainted with the Distributed Computing Environment, read Chapters 1 and 2. To evaluate z/OS DCE for your organization, read chapters 3 and 5. Access to CICS® and IMS™ programs and data with the Application Support servers is described in Chapter 4.

---

## Where to Find More Information

Where necessary, this book references information in other books using shortened versions of the book title. For complete titles and order numbers of the books for all products that are part of z/OS, see the *z/OS Information Roadmap*, SA22-7500. For complete titles and order numbers of the books for z/OS DCE, refer to the publications listed in the “Bibliography” on page 37.

For information about installing z/OS DCE components, see the *z/OS Program Directory*.

## Softcopy Publications

The z/OS DCE library is available on a CD-ROM, *z/OS Collection*, SK3T-4269. The CD-ROM online library collection is a set of unlicensed books for z/OS and related products that includes the IBM Library Reader.™ This is a program that enables you to view the BookManager® files. This CD-ROM also contains the Portable Document Format (PDF) files. You can view or print these files with the Adobe Acrobat reader.

## Internet Sources

The Softcopy z/OS publications are also available for web-browsing and for viewing or printing PDFs using the following URL:

<http://www.ibm.com/servers/eserver/zseries/zos/bkserv/>

You can also provide comments about this book and any other z/OS documentation by visiting that URL. Your feedback is important in helping to provide the most accurate and high-quality information.

## Using LookAt to Look up Message Explanations

LookAt is an online facility that allows you to look up explanations for z/OS messages. You can also use LookAt to look up explanations of system abends.

Using LookAt to find information is faster than a conventional search because LookAt goes directly to the explanation.

LookAt can be accessed from the Internet or from a TSO command line.

You can use LookAt on the Internet at:

<http://www.ibm.com/servers/eserver/zseries/zos/bkserv/lookat/lookat.html>

To use LookAt as a TSO command, LookAt must be installed on your host system. You can obtain the LookAt code for TSO from the LookAt Web site by clicking on the **News and Help** link or from the *z/OS Collection*, SK3T-4269.

To find a message explanation from a TSO command line, simply enter: **lookat** *message-id* as in the following:

```
lookat iec192i
```

This results in direct access to the message explanation for message IEC192I.

To find a message explanation from the LookAt Web site, simply enter the message ID and select the release with which you are working.

**Note:** Some messages have information in more than one book. For example, IEC192I has routing and descriptor codes listed in *z/OS MVS Routing and Descriptor Codes*, SA22-7624. For such messages, LookAt prompts you to choose which book to open.

## Accessing Licensed Books on the Web

z/OS licensed documentation in PDF format is available on the Internet at the IBM Resource Link site:

<http://www.ibm.com/servers/resourceLink>

Licensed books are available only to customers with a z/OS license. Access to these books requires an IBM Resource Link user ID, password, and z/OS licensed book key code. The z/OS order that you received provides a memo that includes your key code.

To obtain your IBM Resource Link user ID and password, logon to:

<http://www.ibm.com/servers/resourceLink>

To register for access to the z/OS licensed books:

1. Logon to Resource Link using your Resource Link user ID and password.
2. Select **User Profiles** located on the left-hand navigation bar.
3. Select **Access Profile**.
4. Select **Request Access to Licensed books**.
5. Supply your key code where requested and select the **Submit** button.

If you supplied the correct key code you will receive confirmation that your request is being processed.

After your request is processed you will receive an e-mail confirmation.

**Note:** You cannot access the z/OS licensed books unless you have registered for access to them and received an e-mail confirmation informing you that your request has been processed.

To access the licensed books:

1. Logon to Resource Link using your Resource Link user ID and password.
2. Select **Library**.
3. Select **zSeries**.
4. Select **Software**.
5. Select **z/OS**.
6. Access the licensed book by selecting the appropriate element.

---

# Chapter 1. Distributed Computing

Distributed computing is built upon the principle of functional distribution. This means that not all processing functions are performed by a single system; rather, they are distributed to multiple systems, allowing for the exploitation of each system's specialized services or computing strengths.

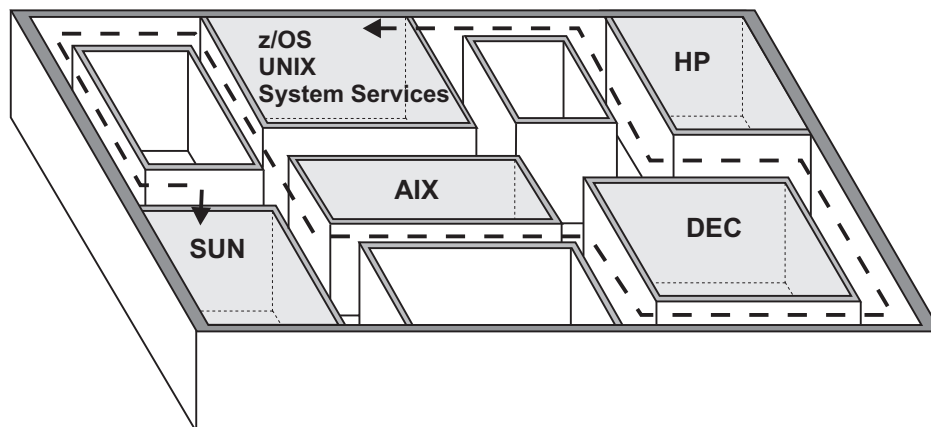
---

## The Need

Traditionally, computers have been independent centralized systems, consisting of a central processor connected to peripherals such as storage units, display monitors, and other input and output devices. Although they were appealing in terms of cost and control, these systems often lacked responsiveness. The people who used them, typically in finance, engineering, and sales, often had conflicting needs. The compromises made to balance those needs often left no user completely happy with the results. Each group of users soon acquired what industry author Grayce Booth describes as the "If only we could have our own computer" syndrome.

The installation of inexpensive, stand-alone, departmental minicomputers and workstations was an attempt to return responsive computing to the user community. But as the number of computer systems in the office grew, their users soon realized they needed to share data and resources among the individual computers. It was no longer feasible to have truly independent systems; the computers needed to work together.

The biggest obstacle to computer cooperation has been the lack of compatibility between systems. The myriad of operating systems, formats, standards, and protocols has made it almost impossible for different systems to communicate and share data. As Figure 1 shows, systems can be connected, but this linking can be a difficult path through a complex maze.



*Figure 1. A Typical Complex Computing Environment*

The larger the number of different operating systems, application environments, and network protocols, the more complicated and convoluted is the route that ties these entities together.

---

## The Solution

A distributed computing environment (DCE) allows several individual systems to share data, processors, applications, and devices and to operate as a single unit. It also allows the individual units that comprise the distributed environment to be on different vendor platforms and running different operating systems. The participating machines can range from personal computers to supercomputers; the network can connect machines in one building or on different continents.

Plugging a distributed computing environment into this maze opens the doors between the independent systems so they can work cooperatively.

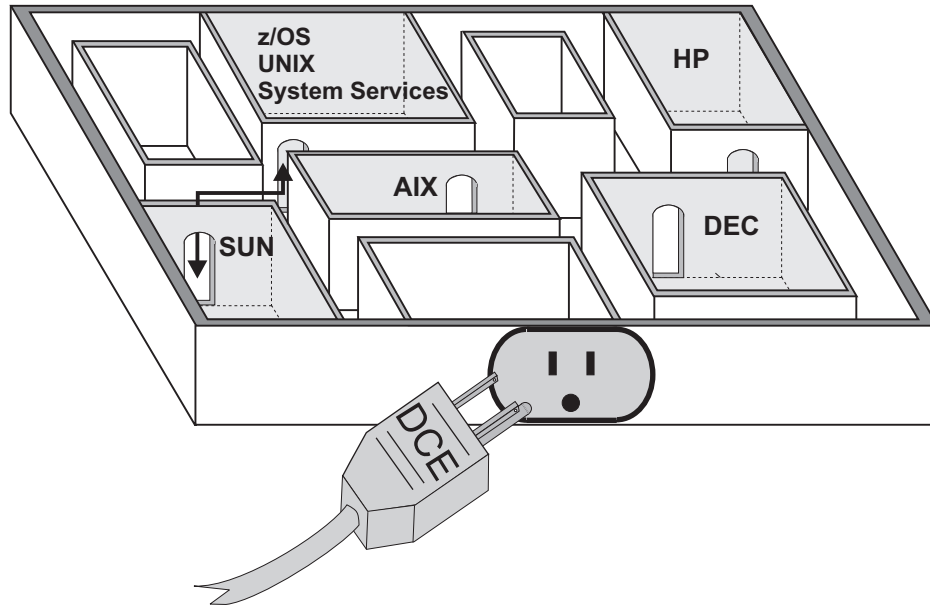


Figure 2. Making the Connection with a Distributed Computing Environment

In a distributed computing environment, two or more computers are linked together over a network in a way that allows them to share resources, programs, and devices. System connection no longer needs to be a complex tangle of protocols, operating systems, and applications.

---

## The Benefits

From a business point of view, distributed computing offers greater flexibility. When you can link systems from different vendors, you have more choices in the design of your data processing operations. Growth and change can occur more naturally, based on evolving individual and departmental needs rather than a centrally imposed plan. Computers that are already in place today can be tied together to form a new, enterprise-wide or global system, preserving existing investments in equipment and data. Pieces of the network that share data or function can serve as backups if other parts of the distributed environment become unavailable.

Even though the distributed system can be composed of many different pieces potentially located miles apart, the user sees the network and all its hardware and software components as a single common system. The end user, who, for example, wants to monitor sales, is unaware of the underlying complexities of her computing environment. Figure 3 shows her printing a sales report. All she needs to know is how to use her menus to manipulate the sales information and print out reports.

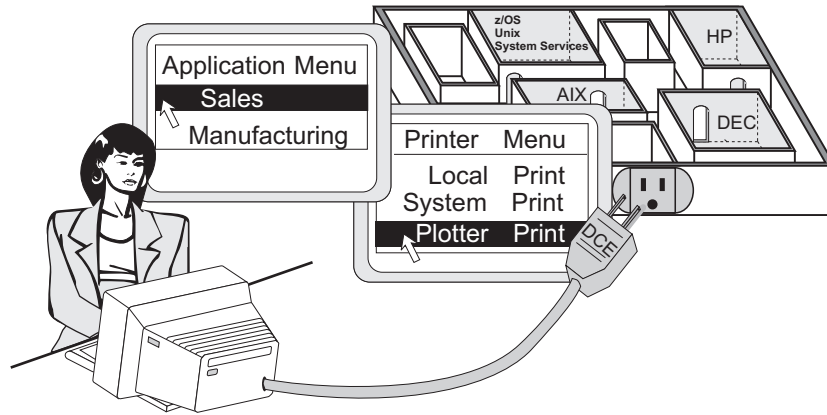


Figure 3. End User's View

A distributed computing environment gives the end user maximum flexibility with minimal explicit knowledge of the environment. A simple end user view is the user's connection to the distributed computing environment.

---

## Open Systems

Open Systems is an integral part of any distributed computing environment. Open Systems refers to both a technology and a philosophy for enabling interoperability between different types of computer systems. As a technology, Open Systems means adhering to international standards that allow the interoperability and portability of data, applications, and skills across computer platforms. Philosophically, Open Systems means that computer vendors are moving away from closed proprietary technologies and toward an open cooperative environment.

## The Open Software Foundation

Many of the key players in the computer industry have been working to address the widely recognized need for Open Systems. A number of these companies, including IBM, plus several research and development organizations, have joined the Open Software Foundation (OSF). OSF is a nonprofit partnership whose sole purpose is to make the computer marketplace an Open Systems environment. The vehicle for delivering this computing environment is OSF Distributed Computing Environment (OSF DCE).

To accomplish its task, OSF issued an invitation to the members of the computer industry worldwide to submit existing technology components for inclusion in OSF DCE. A major benefit of OSF DCE is that it is based on established running code. It is not merely the specification of an evolving standard; it is a present day solution.

DCE provides a high-level, coherent environment for developing and running applications on a distributed system. DCE provides tools for developing distributed applications and services for running applications.

DCE is oriented toward heterogeneous systems. Although a distributed system can be implemented using a single operating system running on all nodes in a network, the DCE architecture allows for different operating systems and different hardware platforms. Applications built using DCE can be ported to other platforms running DCE software.

DCE supports the sharing of data and function throughout the distributed system. Authorized users can access data regardless of their operating system, hardware platform, or location in the distributed system.

## IBM and DCE

DCE is strategic for IBM customers who want open system solutions.

z/OS DCE is part of IBM's open and distributed strategy for z/OS. This environment, called z/OS UNIX System Services, is a cohesive environment where the applications in the information network:

- Exploit the strengths of z/OS, using an integrated set of open system services
- Exchange data and access services transparently
- Are easily portable across multivendor operating systems that support open standards
- Are managed, along with their data, across the network.

z/OS combines DCE components and other interoperability-enabling services, forming a comprehensive, integrated, and all-encompassing structure to handle every aspect of distributed computing, from communications to resource management to application development. Figure 4 highlights the key interoperability components of z/OS UNIX System Services:

- DCE Remote Procedure Call (RPC) allows access to procedures located anywhere in the network.
- DCE Directory provides a consistent view of distributed system resources and services.
- DCE Time coordinates time values in a distributed network.
- DCE Security, including access control, identification and authentication, provides trustworthy identification of users, secure communications, and controlled access to resources.
- Conversational support is available with Advanced Program-to-Program Communication (APPC). APPC provides common interprogram communication services, common logical definition for peer connectivity and hierarchical connectivity, and data stream independence.
- Messaging and Queuing Interface is the model for interprogram communication for CICS/ESA® and IMS/ESA®.



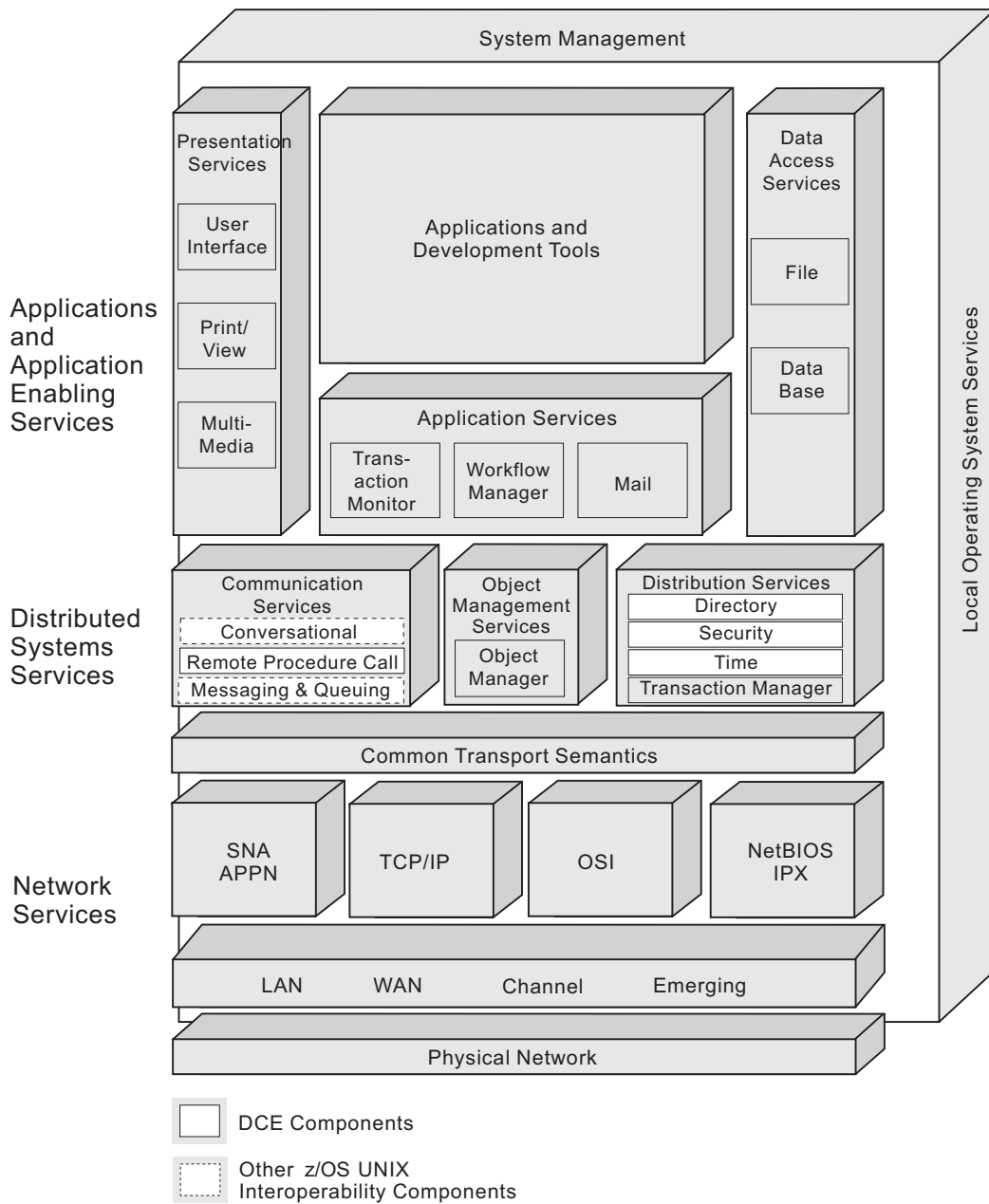


Figure 4. The Interoperability Components

As you read the following chapters, you will find out more about DCE, its components, benefits, and uses. You will find answers to questions that you may have about:

- System reliability
- Ease of use
- Security
- Responsiveness
- Flexibility
- Economic feasibility.

---

## Chapter 2. DCE Features

This chapter describes the DCE services and how they relate to each other. It also defines and explains key terms.

To operate effectively and benefit your organization, distributed computing must achieve the following goals:

- Provide the tools and services that support the development of distributed applications
- Ensure that the support services are integrated and comprehensive
- Provide for the interoperability and portability of applications across heterogeneous platforms
- Support data sharing throughout the system
- Participate in the global computing environment by being compatible with other computing standards (an example is X.500).

---

### The Client/Server Model

A distributed computing environment links computers together so they can share resources, programs, and devices. Applications and data can be distributed among the individual machines that make up the distributed environment. This system connection is based on the client/server model.

In the client/server model, the distributed application is divided into two parts: client and server. The client side is a machine or process that initiates a request and receives the benefit of a specialized service. The server side is a machine or process that provides the requested specialized service.

The terms client and server are relative roles. For example, the Raj-man Music Corporation, a fictitious record company, wants its monthly newsletter to include an article listing its top selling compact discs. The list will include the title, artist, quantity sold, and gross revenue generated. As illustrated in Figure 5, three systems are involved in creating the newsletter: publishing, inventory, and accounting. The publishing system, acting as a client, requests the required monthly sales information from the inventory system. The inventory database machine, in its role of server, can determine the top sellers providing the required information on artist, title, and quantity, but it does not contain the revenue details. The inventory server becomes a client, requesting pricing information from the accounting system. Upon receipt of that information, the database can calculate the gross revenue, and return all the required information to the publishing system for printing.

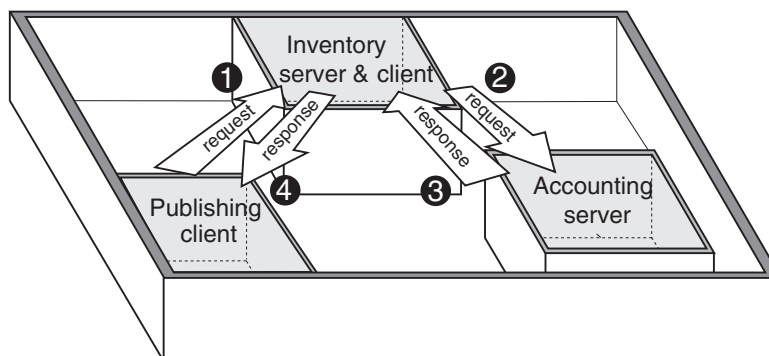


Figure 5. Client/Server Communication

In general, a server node is specialized. It performs a function that is only found on that particular machine. Client nodes are generalized. Typically, many nodes can be clients of a specialized service, and these nodes can also be clients for many different services.

## Overview of DCE

The newsletter scenario also illustrates the complexities of a distributed environment. As illustrated in Figure 6, it is quite probable that the three different computers involved are from three different manufacturers, each with a different operating system. The newsletter might be created on a Sun workstation, using its desktop publishing capabilities. z/OS, with its large database and data integrity strengths, could be used to maintain the inventory information. And accounting could be done on a departmental DEC machine.

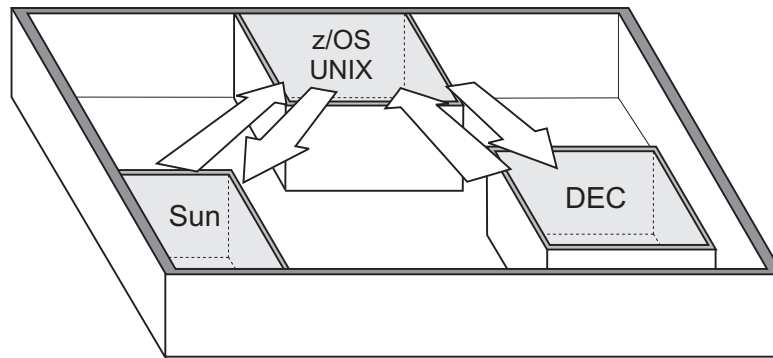


Figure 6. Complex System Connections

The complexities of such an environment make it difficult to link resources from an application standpoint. Programmers can write applications that cooperate across machines by explicitly writing the code that performs the network communications, but this task requires much time, expertise, and expense. Wherever the differences between systems or network protocols show through to the application programmer, the opportunity for failure is increased.

DCE provides the connection between the operating system and network on one hand and the distributed application on the other. Figure 7 shows how DCE provides the services that allow distributed applications to plug into a collection of heterogeneous or homogeneous computers, operating systems, and networks as if they were a single system.

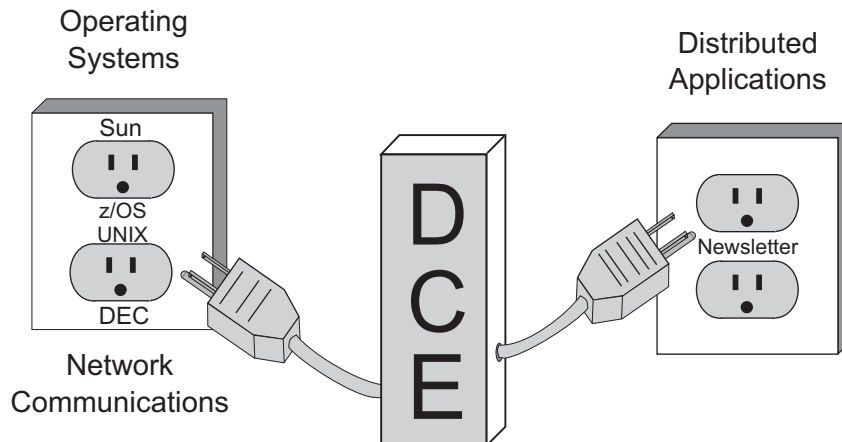


Figure 7. The DCE Connection

Several technology components work together to implement the DCE connection.

## DCE Architecture

DCE technologies fall into two general categories: programming services and distributed services.

The DCE Threads Service and DCE Remote Procedure Call (RPC) are programming services; they include libraries that implement application program interfaces (APIs) and program development tools.

The remaining DCE technologies are distributed services. Each consists of a daemon, or server process, that runs continuously on a machine and responds to requests sent over the network. These services have administrative components to manage the service, and have APIs which programmers can use to access these components.

Application programmers deal mostly with the programming services. Although the distributed services are accessed through APIs, the programmer usually only uses them indirectly through RPC, which in turn uses the APIs of the distributed services. Administrators, on the other hand, deal mostly with the distributed services, because these have significant management requirements.

Figure 8 shows the DCE architecture and its technology components, along with their relationship to applications, underlying system support, and placeholders for future technologies.

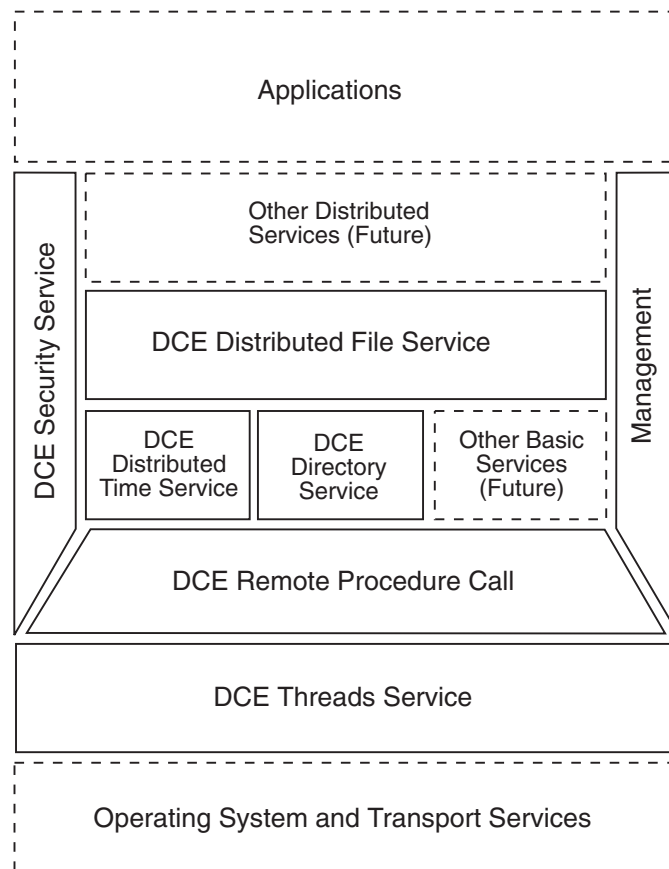


Figure 8. The OSF DCE Architecture

The following sections briefly describe DCE services.

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

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

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

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

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

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

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

**Distributed Time Service:** The DCE Distributed Time Service (DTS) provides synchronized time on the computers participating in a distributed computing environment. In a single system, one clock provides the time to all applications. In a distributed system, however, each node has its own clock. Even if all the clocks in a distributed system could be set to one consistent time at some point, they would drift away from that time at different rates. As a result, different nodes would have different times. This is a problem for distributed applications where the ordering of events is important. DTS enables distributed applications to determine event sequencing, duration, and scheduling. DTS synchronizes a DCE host's time with Universal Coordinated Time (UTC), an international time standard.

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

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

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

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

**Interoperability between RACF® and z/OS DCE:** z/OS DCE also provides interoperability between Resource Access Control Facility (RACF) on z/OS and z/OS DCE.

**Note:** (RACF is a component of the SecureWay® Security Server for z/OS.) RACF is an external security manager (ESM). ESMs other than RACF may be used with the interoperability feature of z/OS DCE if they have the same function as RACF.

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

- Appropriately authorized DCE servers to acquire corresponding z/OS security credentials for the DCE client and use the DCE client's corresponding z/OS user ID for access to RACF-authorized resources.
- A z/OS 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 z/OS and can invoke a DCE program without reauthenticating to DCE.

## Integrating the DCE Components

Figure 9 on page 12 shows how the DCE components work together in a distributed environment.

1. Distributed Time Service (DTS) keeps the clocks on the different machines synchronized.
2. Each server is registered with the Directory Service. The server processes (Procedure X and Procedure Y) advertise their locations (host address and network identification) in the Directory.
3. The client's RPC queries the Directory for a server offering Procedure X which is required or called in Application 1.
4. The Security Service verifies the identity and authority of both the client and server.
5. The client (Application 1) makes a remote procedure call to the server for Procedure X.
6. The server checks the client's security credentials.
7. The server performs Procedure X.
8. The server returns the results to the client application.

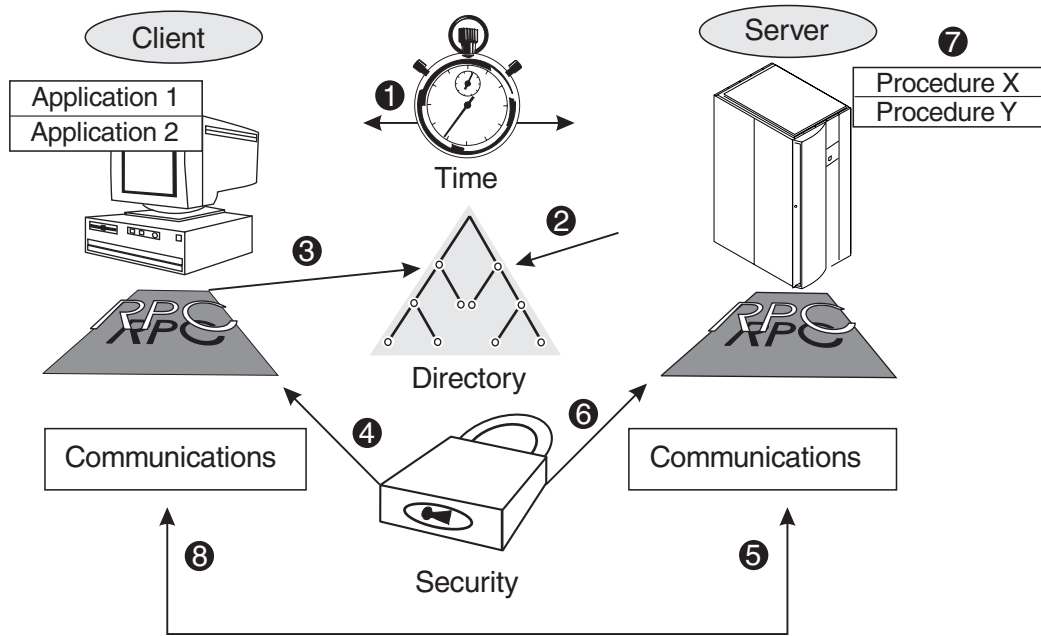


Figure 9. DCE's Integrated Components

The DCE components, although running in different locations, are so well integrated that users do not know or need to know if the applications are running locally (on their machine) or remotely.

This integration also hides the details of security and directory from the application programmers using them. Applications written on one machine can run on other machines, the location of required or called procedures is handled by the directory. The portability of the code, the hidden details of security and the directory, and the transparency of the communications interface ensures ease and flexibility for programmers who are creating distributed application programs.



## Chapter 3. z/OS DCE

The implementation of the OSF DCE technology is a continuation of IBM's strategy of open systems and interoperability across different platforms. This open and distributed strategy views z/OS both in the roles of server to other client systems and as a peer with other server systems in the network.

### Exploiting System Strengths

DCE promises the freedom to design a network of resources best suited to your present needs—in terms of capability, performance, and cost—with assurance that the current investment in applications and data will be portable to the new configurations of the future. z/OS DCE combines the strengths of z/OS and the flexibility of open systems into a distributed computing environment. Figure 10 demonstrates that the traditional strengths of z/OS can be combined with business solutions available on other platforms, whether these be other IBM operating systems or those from other vendors.

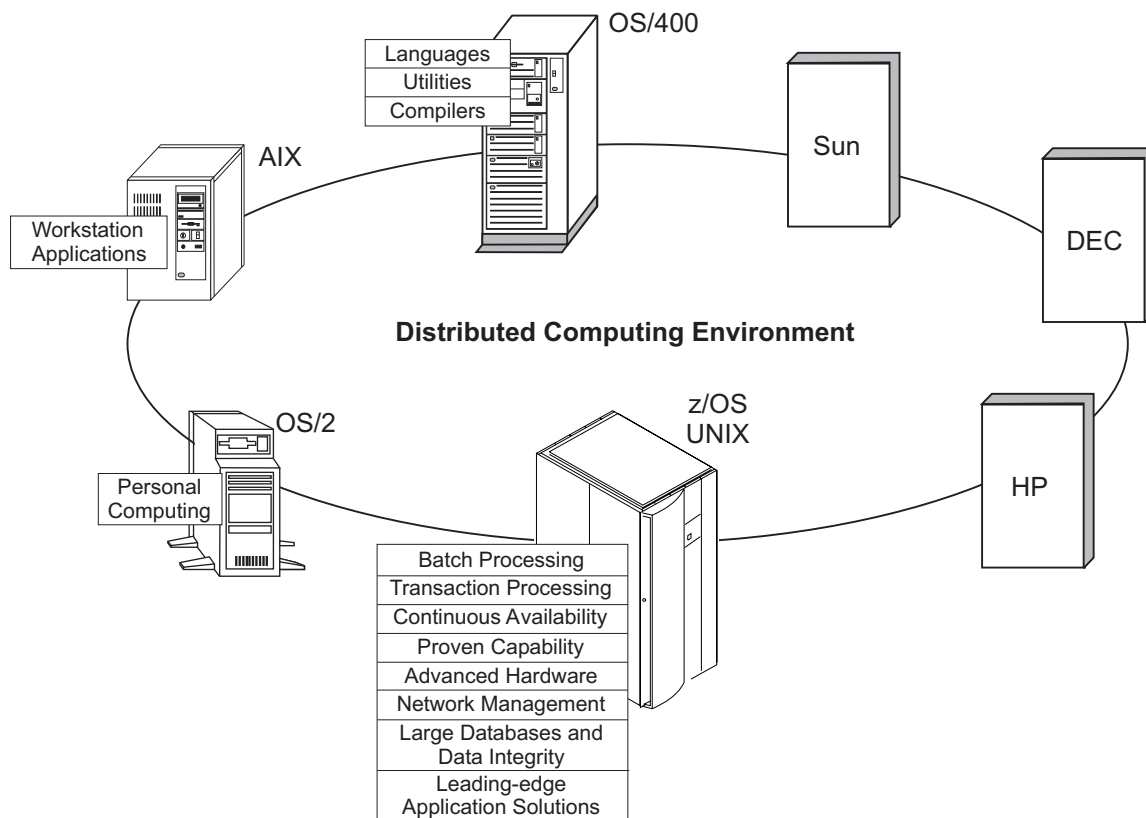


Figure 10. z/OS in a Distributed Computing Environment

With z/OS DCE, you can build an environment of larger operating systems or servers and of distributed systems and workstations that share common interfaces. Your enterprise can:

- Make your z/OS applications and data available to users on other systems.
- Create client/server applications developed on workstations that can be ported to other systems, such as z/OS, in your distributed computing environment.
- Maintain the industrial strength of z/OS while gaining access to applications and data in multivendor systems.

Your enterprise can use the traditional capabilities of z/OS as a powerful platform in an open and expanding environment of computing resources.

## Existing Cooperative Processing Applications

Today, many z/OS-based enterprises, requiring fast and accurate exchanges of data, have developed client/server applications to address their data processing and communications needs. These applications, sometimes called cooperative processing applications, exploit the strengths of other computers using IBM's Advanced Program-to-Program Communications (APPC) in their MVS/ESA™ environments. APPC is an implementation of the Systems Network Architecture (SNA) LU 6.2 protocol. APPC allows interconnected systems to communicate and share the processing of programs. APPC provides a set of z/OS callable services that enable z/OS application programs to communicate with other application programs through communication protocols provided by the SNA network.

z/OS DCE protects an enterprise's investment in APPC-based cooperative applications. APPC-based applications and RPC-based applications can coexist in your enterprise.

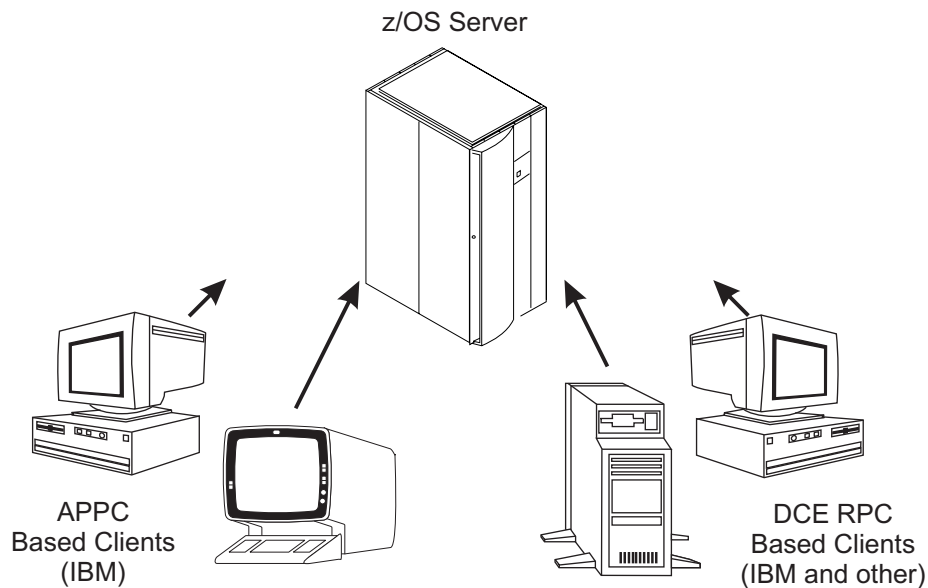


Figure 11. z/OS DCE Client/Server Support

## Using DCE for Online Transaction Processing (OLTP)

The z/OS Encina Toolkit Executive enhances DCE with transactional semantics. Encina is a family of products that provides a range of services for developing and deploying large scale client-server and open, distributed transaction processing systems. Layered on top of OSF DCE, Encina uniquely extends DCE with services such as transactions that are necessary when building business-critical systems. It implements a two-phase commit protocol. This is a set of actions that ensures an application program makes all the changes of a transaction to a collection of resources or makes no changes. An Encina client program can be a standalone z/OS application or part of an IMS transaction program that also updates IMS data. The z/OS Encina Toolkit Executive is a subset of Transarc's Encina.

## Accessing CICS and IMS

DCE access to Customer Information Control System (CICS) and Information Management System (IMS) programs and data is available with the optional z/OS DCE Application Support servers. The Application Support servers allow an enterprise to extend the strengths of the CICS and IMS processing and development beyond the z/OS environment and into the world of open systems. Chapter 4, “Application Support Servers” on page 19 describes the z/OS DCE Application Support.

---

## z/OS DCE Architecture

Figure 12 on page 16 summarizes the services available with z/OS DCE. These include:

- The base DCE services:
  - Remote Procedure Call
  - Directory Service
  - Distributed Time Service
  - Threads Service
  - Security Service
- Distributed File Service (DFS)
- z/OS DCE Application Support CICS and IMS

Available as an optional feature of z/OS DCE:

- z/OS SecureWay Security Server DCE
- z/OS Encina Toolkit Executive

**Note:** In this book the term “DCE Security Server” (or simply “Security Server”) refers to the z/OS SecureWay Security Server DCE or to a DCE Security Server provided on another host in the DCE cell. The z/OS SecureWay Security Server DCE is a component of the SecureWay Security Server for z/OS.

This selection of services enables a closer coupling of IBM services, as well as enabling IBM systems to function as parts of a heterogeneous distributed system.

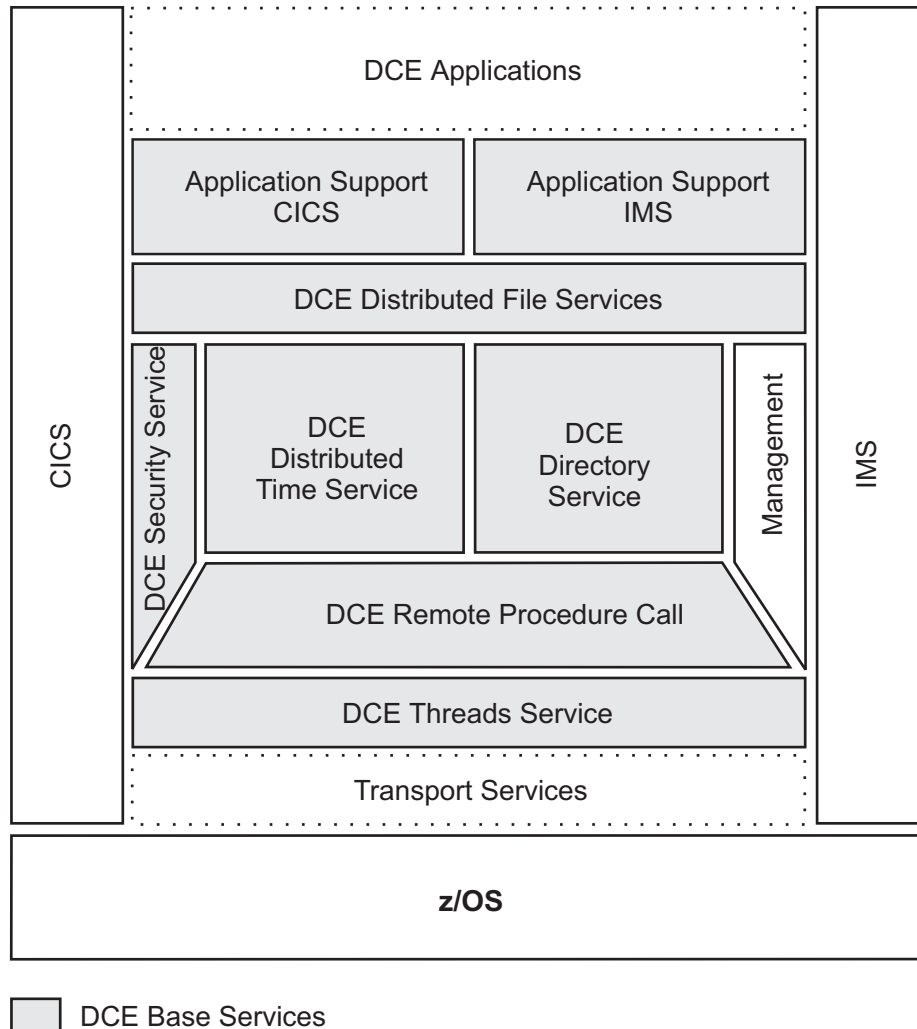


Figure 12. z/OS DCE Architecture

## DCE Participation

With z/OS DCE, z/OS can participate in single-cell and multi-cell configurations of DCE. A cell is DCE's basic unit of organization. It usually consists of users, machines, and resources sharing a common purpose. Members of an organization who are working on the same project are likely to belong to the same cell. In a large organization with several cells, the sales team could belong to one cell and the finance team could belong to a second cell. A small organization might only have one cell for both sales and finance, because they share the same financial information and the same level of security. Members of a cell are usually located in a common geographic area, but they can be located in different buildings, different cities, or even different countries, providing good connectivity exists.

As illustrated in Figure 13 on page 17, a z/OS machine can be a server (or client) to machines in the same cell or other DCE cells. RPC is used for communication between cells.

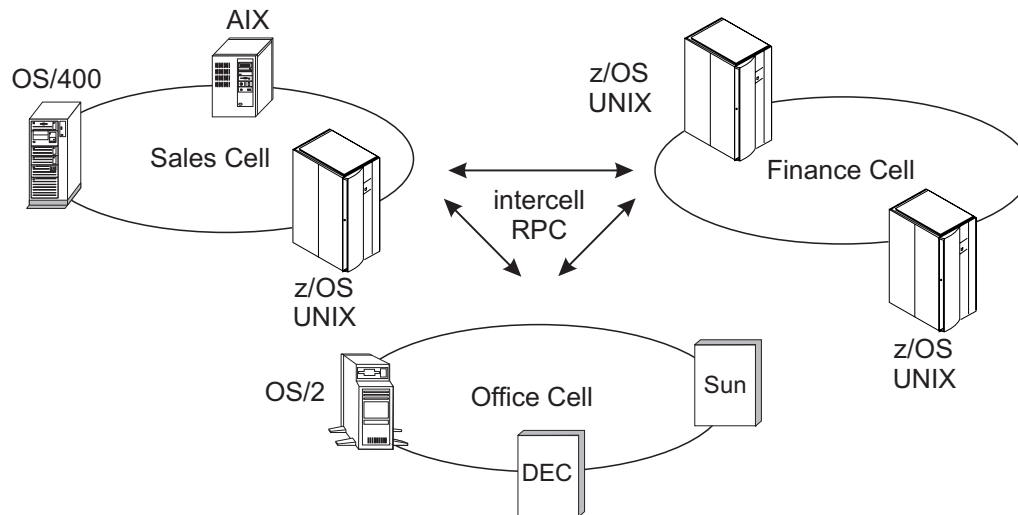


Figure 13. z/OS Cell Participation

Cells can contain more than one z/OS machine, and z/OS machines can be clients and servers of each other.



## Chapter 4. Application Support Servers

z/OS DCE Application Support is designed to increase the availability of data throughout an enterprise. z/OS DCE facilitates distributed access to Customer Information Control System (CICS) and Information Management System (IMS) programs and data. Users throughout the Distributed Computing Environment can exploit the strengths of CICS and IMS, with expanded access to these transaction processing and development environments. Client applications anywhere in the DCE environment can access CICS and IMS, including the data contained in these environments, for example DB2® and DL/I.

Other products on the market today allow workstation access to CICS and IMS, but they use communication gateways to connect different network or system architectures. These gateways only allow data interchange between the workstation and the CICS or IMS systems on the specified z/OS host. Because the Application Support servers are specialized DCE servers, they can offer the heterogeneous system interoperability only available in a distributed computing environment. The same workstation code can be used to access CICS or IMS data and transactions on all z/OS hosts in the distributed computing environment. The client code is simply an RPC routine, not a special gateway conversation.

Figure 14 illustrates transparent access to the RPC client through z/OS DCE Application Support CICS and IMS.

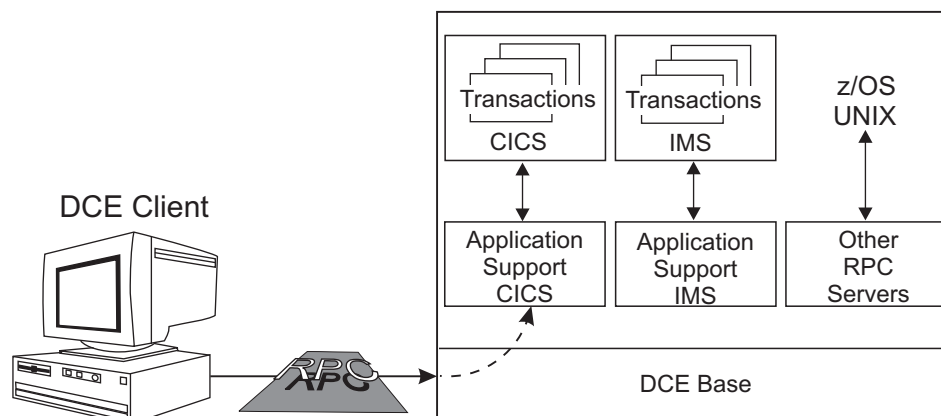


Figure 14. Application Support server

z/OS DCE Application Support provides the capability of workstation integration. The DCE client can be non-CICS, non-IMS, or non-IBM. The client can access multiple CICS and IMS regions and other z/OS and non-z/OS applications and data at the same time. The Application Support servers allow you to:

- Continue to use existing host applications
- Extend CICS and IMS support to your workstations
- Write new client applications using the DCE client/server model
- Write new transactions in CICS which are available as DCE servers
- Provide the workstation user with a consistent interface to z/OS and non-z/OS DCE servers, and CICS and IMS transactions.

The Application Support servers not only protect the z/OS customers' investment in applications and data residing within CICS and IMS, but also expands access to these transaction processing and development environments.

## **Application Support Transactional RPCs for IMS and CICS**

Application Support RPC support for IMS and CICS transactions lets Application Support servers participate in Encina distributed transaction processing applications. The Application Support provides transactional RPC support for IMS and CICS transactions. For consistent transformation of data, all operations are performed as a single unit of work to become permanent or are undone.



---

## Chapter 5. Working in a Distributed Computing Environment

If your organization has enjoyed the productivity of z/OS, z/OS DCE can offer you the additional productivity of distributed computing. You can benefit from DCE if you have:

- Departments with isolated computing resources that need to be connected with computers in other environments.
- Personal computers that need the computational and storage facilities only available on large machines.
- Computers from different manufacturers that need to communicate with each other.
- Services that would be more efficient if the processing was split among two or more machines, exploiting the computing strengths of each machine.

---

### Using DCE

z/OS DCE expands your computing environment into the world of distributed heterogeneous computing. In this expanded environment:

- End users can continue to exploit the strengths of z/OS UNIX System Services and at the same time be part of a larger computing environment where applications and data are shared.
- Application programmers can write programs that are portable across platforms and that can serve users and access data without the user or the programmer having to know where the applications and data are located.
- System administrators must know the network configuration, but the Directory and Security Services provide the tools necessary to facilitate administration without affecting application programs. Administration information can be distributed and replicated as needed.

Figure 15 illustrates the different user views of z/OS DCE.

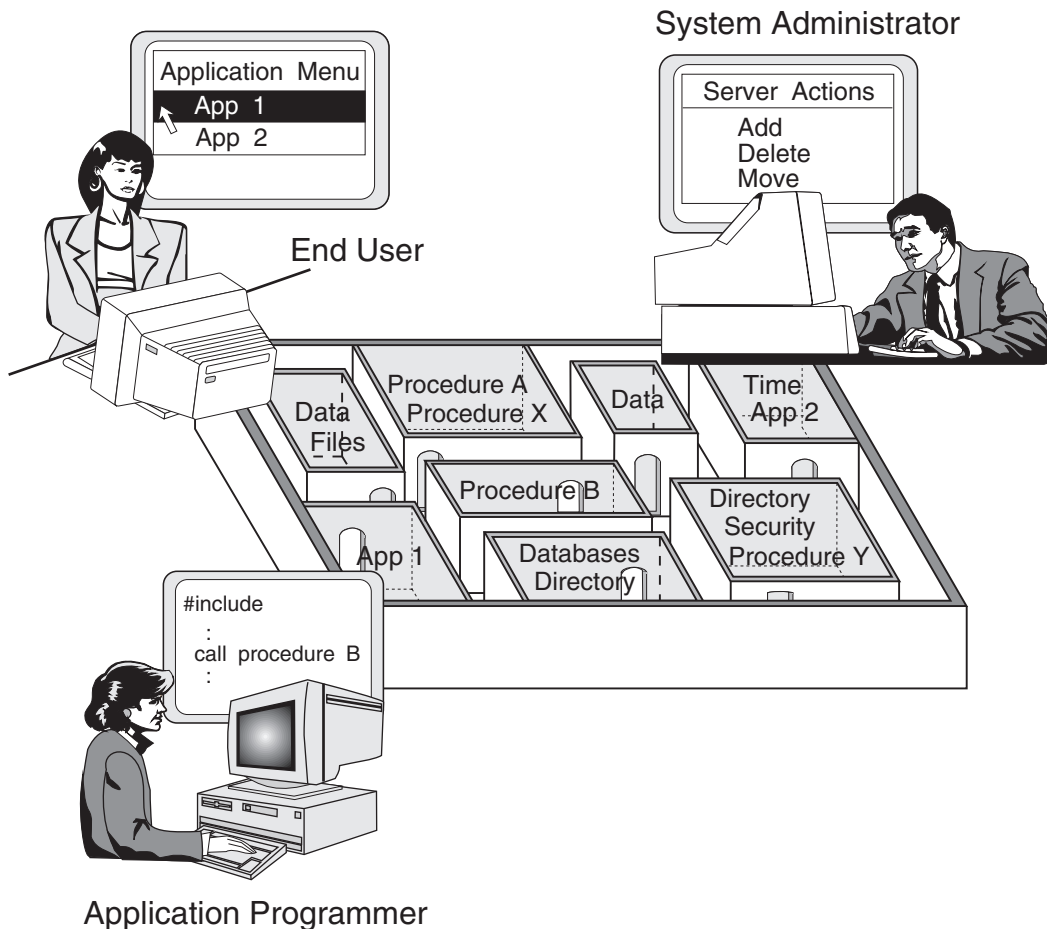


Figure 15. Views of z/OS DCE

Although their roles are expanded, z/OS DCE users are not aware of the complexities of the underlying distributed environment.

- The end user has Application 1 (App 1) running on her machine. She does not need to know any of the processing required to run the application. A simple menu is her connection to the Distributed Computing Environment.
- The application programmer views the resources in a logical way. She uses Procedure B in her program, but she does not need to know where it is located. She only needs to provide the appropriate call to access that code.
- The system administrator needs a more detailed knowledge of the environment. He needs to know what the resources are, where they are located, and how the networks are configured. But he only needs to enter this information once, then he can replicate and distribute it as required. He can also manage multiple systems from one location using the facilities provided with the Directory and Security Services.

Figure 15 also illustrates how the DCE Services can be distributed throughout your environment. A single machine can provide a number of functions. Depending on the importance of availability and performance, you might choose to duplicate data and functions on other machines.

---

## **z/OS DCE Scenarios**

The following scenarios describe sample z/OS DCE implementations for fictitious organizations. The first scenario involves a manufacturing company, while the second uses an educational example. Though these two environments differ greatly, users in each had similar concerns when it came to implementing a distributed computing environment:

- Is the system reliable?
- Is it easy to use?
- Is it secure?
- Is it responsive?
- Is it flexible?
- Is it economically feasible?

### **In a Manufacturing Environment**

Founded in 1958, Leanside Industries designs and builds ergonomic office equipment. Its history with computers is somewhat typical. In the mid-1960's, users of the centralized computer system felt that the operations staff did not respond to their needs. As a result, any department that could justify the expenditure acquired its own computer. Separate computers supported engineering, accounting, and manufacturing. Processing capabilities were spread throughout the organization.

This distributed data processing had certain advantages:

- Reduced load on the host  
This unburdened overworked central computers.
- Quicker response time  
Users no longer had to wait for information to be processed and then transmitted back and forth on the communication lines.
- More tailored applications  
Users now had access to applications tailored for local use, rather than more generalized host routines.

But there were also some major disadvantages:

- Software and memory limitations  
Local processors did not handle sophisticated programs well.
- Maintaining program consistency  
Changes in one department's program needed to be made in all other areas that used copies of the same program.
- Equipment incompatibility  
Minicomputers, modems, and processors from many different vendors could not always be connected.
- Complexity  
Not only was there a multiplicity of vendors, equipment, and programs, there was also a multiplicity of people and places. In a centralized system, these pieces were under strict control. In the new decentralized environment there was a perceived loss of control.

Figure 16 on page 24 illustrates Leanside's current environment. The engineering and manufacturing departments are doing most of their processing locally, but they are also using host connected

nonprogrammable terminals and workstation terminal emulation programs to maintain computing connections with the central processor. The accounting department does not have its own machine and does all its work on the central processor. There are no computing links between the departments.

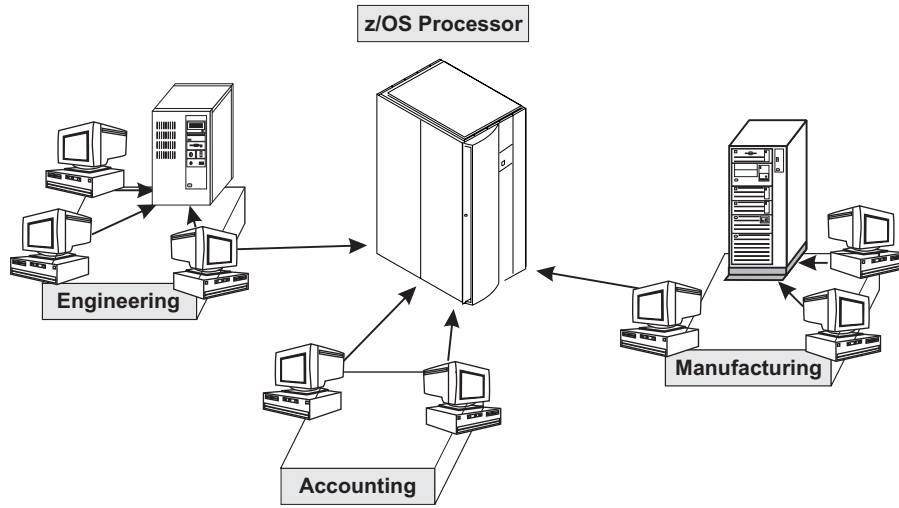


Figure 16. Leanside's Existing Environment

Leanside Industries wants to maintain its departmental computing strengths, but it needs all its computers to work together. All the manufacturers of Leanside's computing equipment are advocates of Open Systems, and each has made DCE available on its various operating systems. This gives Leanside total access to DCE, its departmental systems can share data, processors, applications, and devices and still operate as a single units.

The company wants its employees to easily share design documents, inventory information, and financial data. This common purpose can be best met by the single cell illustrated in Figure 17.

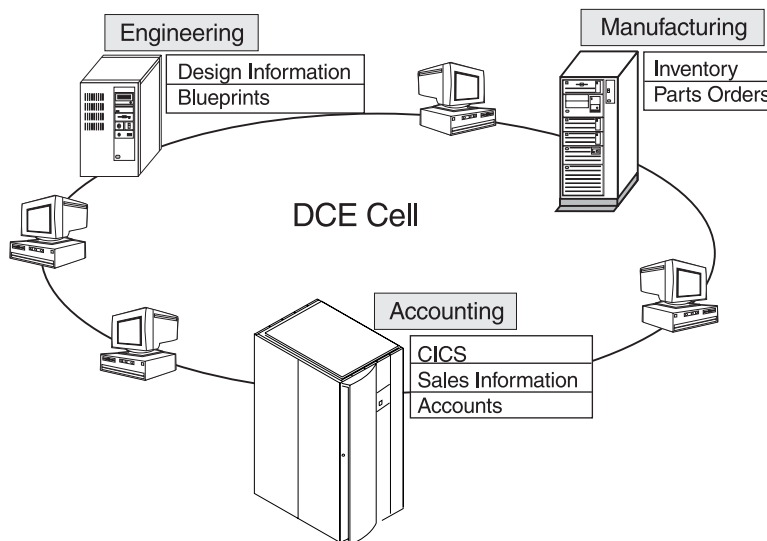


Figure 17. Leanside's New DCE Environment

The cell includes the engineering host with design information, the manufacturing host containing parts inventory, the central host with financial data, and the user workstations. Though not shown in the figure, nonprogrammable terminals gain access to the cell through their host connections. For example, the accounting department can use the cell's resources because the central processor is a member of the DCE cell.

All the hardware components must be connected by a network. Each workstation becomes a DCE client, allowing it to request specialized services. Each host is designated both a DCE server and a DCE client, allowing it to both provide and request services.

Each machine must have the appropriate RPC code installed to enable it to make the necessary client and server connections. The cell requires a time server for time synchronization, a security server for authentication and authorization, and a directory server for application address information. A detailed discussion on the placement of DCE servers can be found in *z/OS DCE Planning*.

Leanside stores its financial data in a CICS database on the central processor. Distributed access to this data is made available through z/OS DCE Application Support.

Leanside Industries has realized significant benefits from its new DCE environment:

- Engineering personnel, who design the office equipment, continue to maintain their documentation as electronic blueprints on their departmental server. DCE Security protects this information from unauthorized users. The engineers can now access the manufacturing database for parts information required during design.
- Manufacturing personnel processing part orders continue to update their inventory, as before, but now this information can be passed immediately to the CICS financial database on the central processor.
- Corporate financial data, including accounts information, billing and sales information, departmental budgetary and expense information, is available throughout the company to those with the appropriate authorization.
- Software and memory limitations have been eliminated. Users have transparent access to machines throughout the environment. As a result, procedures are run on machines best suited for the job, rather than machines local to the user.
- RPC eliminates the need for multiple machines running the same application. With a single machine handling all requests for an application, users can be assured of access to the latest version of all procedures.
- The complexities of the distributed environment are hidden from the end user and the application programmer. DCE gives them greater computational abilities by expanding their access to resources without requiring any in-depth knowledge of behind-the-scenes processing.
- Administration information only needs to be entered once, and then can be replicated and distributed throughout the distributed computing environment.

## In an Educational Environment

The Raj-man Music Corporation, described on page 7, is a founding member of the Music Information Network. This organization is made up of record companies, music publishers, music historians, and college libraries. Its purpose is to provide a music information database for college students specializing in music. Students can access information on composers, artists, styles, and recordings.

Historically, students had been given a list of phone numbers for the various members of the organization. Information gathering involved telephoning the members and asking if they had any information on the student's topic. Many factors affected response time to student requests, including:

- Misunderstandings between the student's request and the member's perception of the request.  
Students sometime complained that they did not get what they asked for. Researchers criticized students for not clearly defining their problems.
- Student requests were filled when the researchers had time.  
Often requests could not be filled in time for student deadlines.

- Research information was mailed to the student.

Students viewed this as an additional delay in their receipt of the information, and on occasion materials never arrived. Member organizations were providing free research service and so were not willing to add courier charges to their expenses.

- There was no way to determine which member was the expert on a particular topic.

Students often found that if they had made just one more phone call they could have gotten exactly what they needed. Members found that they were investing significant research time on topics that they could only provide minimal information.

- Information was being requested by students who were not enrolled in the colleges' music departments.

Researchers did not have the information necessary to determine validity of student requests.

In addition to a common interest in music, all the Music Information Network members had their own computer resources. Each member was using its computers to run its day-to-day business operations, and as a result each had invested in machines best suited to meet its own special requirements. Members realized that the ability to share data and resources among these individual systems would provide huge benefits to the organization. They viewed DCE as the vehicle to link this heterogeneous environment.

Creating the Music Information Network's DCE environment was similar to the steps followed in establishing the Leanside cell:

- A network connects all hardware components.
- Client/server connections are made with RPC.
- Time, Security, and Directory servers are installed.

Unlike the Leanside Industries example, this organization does not have any existing applications that can be ported to the new environment. New applications must be created, and therein lies the Music Information Network's major concern: How to write distributed applications?

DCE application programming requires a logical view of the environment. The programmer needs to consider the placement of data and function, but does not need to understand the physical complexities of the various hardware and software configurations. Most of the effort in developing a DCE application involves the familiar steps of writing and compiling the necessary C code, linking the results with the DCE library and other modules, running the application, and evaluating performance. Before implementing an application, the application programmer writes and compiles an interface definition that defines the application's client/server interface. Although interface definition may be a new task for application programmers, DCE's logical view does not expose them to the underlying system and network protocols involved in distributed computing.

Security can be provided through a set of RPC routines. RPC, in conjunction with the Security Service, can apply authentication and protection to all remote procedure calls.

The new Music Information Network shown in Figure 18 on page 27 gives authorized students direct access to the organization's computers. Now students use distributed applications to query for information. Telephone requests have been eliminated. This online access eliminates time retrieval delays, misunderstanding of requests, and mailing of information. The student provides the application with research keywords, the application then routes the request to the most logical organization member. Fast system response allows the student to quickly determine if the returned information is adequate, allowing additional queries if required. DCE security routines ensure only authorized users can access the information.

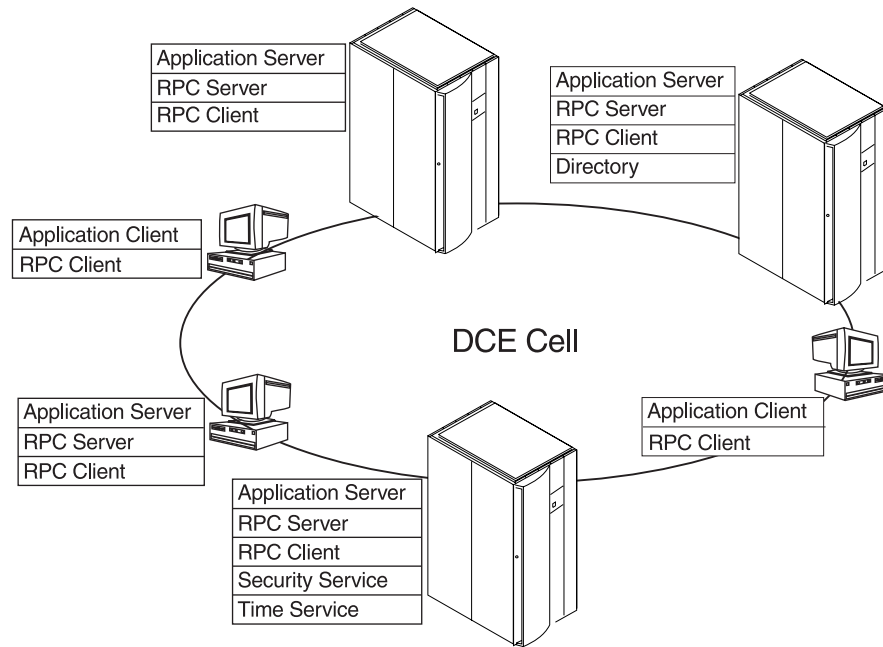


Figure 18. The Music Information Network's DCE Environment

## Summary

z/OS DCE brings z/OS into the heterogeneous distributed computing environment. It is strategic to customers who want open system solutions. It provides the following:

- System Reliability

Incorporating reliability while exploiting the system strengths of the other machines in your environment provides a comprehensive, integrated, and all-encompassing structure to handle all aspects of distributed computing.

- Ease of Use

End users and application programmers are protected from the underlying complexities of z/OS DCE. End users exploit the strengths of z/OS and at the same time can participate in the larger environment. Application programmers can write programs that are portable across multiple platforms.

- Security

DCE provides trustworthy identification of users, clients, servers, and systems. This includes integrity and privacy of communications, and controlled access to resources.

- Responsiveness

DCE allows individual systems to share data, processors, applications, and devices and to operate as a single unit. Application programs can exploit each system's computational strengths without concern for differences in hardware configurations or operating systems.

- Flexibility

DCE gives you the ability to link systems from different vendors offering more choices in the design of your data processing operation. Growth can occur based on evolving individual and departmental needs.

- Economic Feasibility

Computers that are in place today can be tied together to form your new, enterprise-wide or global system, preserving your existing investments in equipment and data.



---

## Appendix A. Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described 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:

IBM Director of Licensing  
IBM Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation  
Licensing  
2-31 Roppongi 3-chome, Minato-ku  
Tokyo 106, Japan

**The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:** INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation  
Mail Station P300  
522 South Road  
Poughkeepsie, NY 12601-5400  
U.S.A

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

---

## Trademarks

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both.

BookManager	CICS	CICS/ESA
DB2	IBM	IBMLink
IMS	IMS/ESA	Library Reader
MVS/ESA	RACF	Resource Link
SecureWay	z/OS	

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.

---

# Glossary

This glossary defines technical terms and abbreviations used in z/OS DCE documentation. If you do not find the term you are looking for, refer to the index of the appropriate z/OS DCE manual or view the *IBM Glossary of Computing Terms*, located at:

<http://www.ibm.com/ibm/terminology>

This glossary includes terms and definitions from:

- *IBM Dictionary of Computing*, SC20-1699.
- *Information Technology—Portable Operating System Interface (POSIX)*, from the POSIX series of standards for applications and user interfaces to open systems, copyrighted by the Institute of Electrical and Electronics Engineers (IEEE).
- *American National Standard Dictionary for Information Systems*, ANSI X3.172-1990, copyright 1990 by the American National Standards Institute (ANSI). Copies may be purchased from the American National Standards Institute, 11 West 42nd Street, New York, New York 10036. Definitions are identified by the symbol (A) after the definition.
- *Information Technology Vocabulary*, developed by Subcommittee 1, Joint Technical Committee 1, of the International Organization for Standardization and the International Electrotechnical Commission (ISO/IEC JTC1.SC1).
- *CCITT Sixth Plenary Assembly Orange Book, Terms and Definitions* and working documents published by the International Telecommunication Union, Geneva, 1978.
- Open Software Foundation (OSF).

The following abbreviations indicate terms that are related to a particular DCE service:

<b>CDS</b>	Cell Directory Service
<b>CICS/ESA®</b>	Customer Information Control System/ESA
<b>DTS</b>	Distributed Time Service
<b>GDS</b>	Global Directory Service
<b>IMS/ESA®</b>	Information Management System/ESA
<b>RPC</b>	Remote Procedure Call
<b>Security</b>	Security Service
<b>Threads</b>	Threads Service
<b>XDS</b>	X/Open Directory Services
<b>XOM</b>	X/Open OSI-Abstract-Data Manipulation

## A

**address.** An unambiguous name, label, or number that identifies the location of a particular entity or service. See *presentation address*.

**APPC.** Advanced Program-to-Program Communication.

**architecture.** (1) The organizational structure of a computer system, including the interrelationships among its hardware and software. (2) The logical structure and operating principles of a computer network. The operating principles of a network include those of services, functions, and protocols.

**authentication.** In computer security, a method used to verify the identity of a principal.

**authorization.** (1) The determination of a principal's permissions with respect to a protected object. (2) The approval of a permission sought by a principal with respect to a protected object.

## B

**binding.** RPC: A relationship between a client and a server involved in a remote procedure call.

**binding handle.** RPC: A reference to a binding. See *binding information*.

**binding information.** RPC: Information about one or more potential bindings, including an RPC protocol sequence, a network address, an endpoint, at least one transfer syntax, and an RPC protocol version number. See *binding*. See also *endpoint*, *network address*, *RPC protocol*, *RPC protocol sequence*, and *transfer syntax*.

## C

**cell.** The basic unit of operation in the distributed computing environment. A cell is a group of users, systems, and resources that are grouped around a common purpose and that share common DCE services.

**Cell Directory Service (CDS).** A DCE component. A distributed replicated database service that stores names and attributes of resources located in a cell. CDS manages a database of information about the resources in a group of machines called a DCE cell.

**client.** A computer or process that accesses the data, services, or resources of another computer or process on the network. Contrast with *server*.

**client context.** RPC: The state within an RPC server generated by a set of remote procedures and maintained across a series of calls for a particular client. See *context handle*. See also *manager*.

**client/server model.** A form of computing where one system, the client, requests something, and another system, the server, responds.

**clock.** The combined hardware interrupt timer and software register that maintains the system time.

**context handle.** RPC: A reference to state (client context) maintained across remote procedure calls by a server on behalf of a client. See *client context*.

**courier.** DTS: A local server that requests a time value from a randomly selected global server. The time value returned is used for synchronization.

**credentials.** Security: A general term for privilege attribute data that has been certified by a trusted privilege certification authority.

\*

**Customer Information Control System (CICS).** An IBM licensed program that enables transactions entered at remote terminals to be processed concurrently by user-written application programs. It includes facilities for building, using, and maintaining databases.

## D

**daemon.** (1) A long-lived process that runs unattended to perform continuous or periodic system-wide functions such as network control. Some daemons are triggered automatically to perform their task; others operate periodically. An example is the **cron** daemon, which periodically performs the tasks listed in the **crontab** file. Many standard dictionaries accept the spelling *demon*. (2) A DCE server process.

**DCE.** Distributed Computing Environment.

**DFS.** Distributed File Service.

**directory.** (1) A logical unit for storing entries under one name (the directory name) in a CDS namespace. Each physical instance of a directory is called a replica. (2) A collection of open systems that cooperates to hold a logical database of information about a set of objects in the real world.

**directory ID.** Directory identifier.

**Directory Service.** A DCE component. The Directory Service is a central repository for information about resources in a distributed system. See *Cell Directory Service* and *Global Directory Service*.

**distributed computing.** A type of computing that allows computers with different hardware and software to be combined on a network, to function as a single computer, and to share the task of processing application programs.

**Distributed Computing Environment (DCE).** A comprehensive, integrated set of services that supports the development, use, and maintenance of distributed applications. DCE is independent of the operating system and network; it provides interoperability and portability across heterogeneous platforms.

**Distributed File Service (DFS).** A DCE component. DFS joins the local file systems of several file server machines making the files equally available to all DFS client machines. DFS allows users to access and share files stored on a file server anywhere in the network, without having to consider the physical location of the file. Files are part of a single, global name space, so that a user can be found anywhere in the network by means of the same name.

**Distributed Time Service (DTS).** A DCE component. It provides a way to synchronize the times on different hosts in a distributed system.

**drift.** DTS: The change in a clock's error rate over a specified period of time.

## E

**endpoint.** RPC: An address of a specific server instance on a host.

**entry.** GDS, XDS: The part of the DIB that contains information relating to a single directory object. Each entry consists of directory attributes.

## G

**Global Directory Service (GDS).** A DCE component. A distributed replicated directory service that provides a global namespace that connects the local DCE cells into one worldwide hierarchy. DCE users can look up a name outside a local cell with GDS.

**group.** (1) RPC: A name service entry that corresponds to one or more RPC servers that offer common RPC interfaces, RPC objects, or both. A group contains the names of the server entries, other groups, or both that are members of the group. See *NSI group attribute*. (2) Security: Data that associates

a named set of principals that can be granted common access rights. See *subject identifier*.

## H

**handle.** RPC: An opaque reference to information. See *binding handle*, *context handle*, *interface handle*, *name service handle*, and *thread handle*.

**heterogeneous.** Pertaining to a collection of dissimilar host computers such as those from different manufacturers. Contrast with *homogeneous*.

**homogeneous.** Pertaining to a collection of similar host computers such as those of one model or one manufacturer. Contrast with *heterogeneous*.

**host ID.** Synonym for *network address*.

## I

**Information Management System (IMS).** A database and data communication system capable of managing complex databases and networks in virtual storage.

**interoperability.** The capability to communicate, execute programs, or transfer data among various functional units in a way that requires the user to have little or no knowledge of the unique characteristics of those units.

**integrity.** RPC: A protection level that may be specified in secure RPC communications to ensure that data transferred between two principals has not been changed in transit.

**interface.** RPC: A shared boundary between two or more functional units, defined by functional characteristics, signal characteristics, or other characteristics, as appropriate. The concept includes the specification of the connection of two devices having different functions. See *RPC interface*.

**interface definition.** RPC: A description of an RPC interface written in the DCE Interface Definition Language (IDL). See *RPC interface*.

**interface handle.** RPC: A reference in code to an interface specification. See *binding handle* and *interface specification*.

**interface specification.** RPC: An opaque data structure that is generated by the DCE IDL compiler from an interface definition. It contains identifying and descriptive information about an RPC interface. See *interface definition*, *interface handle*, and *RPC interface*.

## K

**Kerberos.** The authentication protocol used to carry out DCE private key authentication. Kerberos was developed at the Massachusetts Institute of Technology.

**key.** A value used to encrypt and decrypt data.

## L

**local.** (1) Pertaining to a device directly connected to a system without the use of a communication line. (2) Pertaining to devices that have a direct, physical connection. Contrast with *remote*.

**logical unit (LU).** A host port through which a user gains access to the services of a network.

**LU.** Logical unit.

## M

**manager.** RPC: A set of remote procedures that implement the operations of an RPC interface and that can be dedicated to a given type of object. See also *object* and *RPC interface*.

## N

**name.** GDS, CDS: A construct that singles out a particular (directory) object from all other objects. A name must be unambiguous (denote only one object); however, it need not be unique (be the only name that unambiguously denotes the object).

**name service handle.** RPC: An opaque reference to the context used by the series of next operations called during a specific name service interface (NSI) search or inquiry.

**network.** A collection of data processing products connected by communications lines for exchanging information between stations.

**network address.** An address that identifies a specific host on a network. Synonymous with *host ID*.

**Network Data Representation (NDR).** RPC: The transfer syntax defined by the Network Computing Architecture. See *transfer syntax*.

**network protocol.** A communications protocol from the Network Layer of the Open Systems Interconnection (OSI) network architecture, such as the Internet Protocol (IP).

**node.** (1) An endpoint of a link, or a junction common to two or more links in a network. Nodes can be

preprocessors, controllers, or workstations, and they can vary in routing and other functional capabilities.

(2) In network topology, the point at an end of a branch. It is usually a physical machine.

**NSI group attribute.** RPC: An RPC-defined attribute (NSI attribute) of a name service entry that stores the entry names of the members of an RPC group and identifies the entry as an RPC group. See *group*.

## O

**object.** (1) A data structure that implements some feature and has an associated set of operations. (2) RPC: For RPC applications, anything that an RPC server defines and identifies to its clients using an object Universal Unique Identifier (UUID). An RPC object is often a physical computing resource such as a database, directory, device, or processor. Alternatively, an RPC object can be an abstraction that is meaningful to an application, such as a service or the location of a server. See *object UUID*. (3) XDS: Anything in the world of telecommunications and information processing that can be named and for which the directory information base (DIB) contains information. (4) XOM: Any of the complex information objects created, examined, changed, or destroyed by means of the interface.

**object UUID.** RPC: The Universal Unique Identifier (UUID) that identifies a particular RPC object. A server specifies a distinct object UUID for each of its RPC objects. To access a particular RPC object, a client uses the object UUID to find the server that offers the object. See *object*.

**Open Software Foundation (OSF).** A nonprofit research and development organization set up to encourage the development of solutions that allow computers from different vendors to work together in a true open-system computing environment.

**open system.** A system whose characteristics comply with standards made available throughout the industry and that can be connected to other systems complying with the same standards.

**operation.** (1) GDS: Processing performed within the directory to provide a service, such as a read operation. (2) RPC: The task performed by a routine or procedure that is requested by a remote procedure call.

**organization.** (1) The third field of a subject identifier. (2) Security: Data that associates a named set of users who can be granted common access rights that are usually associated with administrative policy.

**OSF.** Open Software Foundation.

## P

**password.** A secret string of characters shared between a computer system and a user. The user must specify the character string to gain access to the system.

**platform.** The operating system environment in which a program runs.

**port.** (1) Part of an Internet Protocol (IP) address specifying an endpoint. (2) To make the programming changes necessary to allow a program that runs on one type of computer to run on another type of computer.

**position (within a string).** XOM: The ordinal position of one element of a string relative to another.

**position (within an attribute).** XOM: The ordinal position of one value relative to another.

**presentation address.** An unambiguous name that is used to identify a set of presentation service access points. Loosely, it is the network address of an open systems interconnection (OSI) service.

**principal.** Security: An entity that can communicate securely with another entity. In the DCE, principals are represented as entries in the Registry database and include users, servers, computers, and authentication surrogates.

**privacy.** RPC: A protection level that encrypts RPC argument values. in secure RPC communications.

**programming interface.** The supported method through which customer programs request software services. The programming interface consists of a set of callable services provided with the product.

**programming services.** DCE services used mainly by programmers to develop distributed applications. Programming services include RPC and Threads.

**proprietary.** Pertaining to the holding of the exclusive legal rights in making, using, or marketing a product.

**protocol.** A set of semantic and syntactic rules that determines the behavior of functional units in achieving communication.

**protocol sequence.** Synonym for *RPC protocol sequence*.

## R

**RACF.** Resource Access Control Facility.

**remote.** Pertaining to a device, file or system that is accessed by your system through a communications line. Contrast with *local*.

**remote procedure.** RPC: An application procedure located in a separate address space from calling code. See *remote procedure call*.

**remote procedure call.** RPC: A client request to a service provider located anywhere in the network.

**Remote Procedure Call (RPC).** A DCE component. It allows requests from a client program to access a procedure located anywhere in the network.

**request.** A command sent to a server over a connection.

**resource.** Items such as printers, plotters, data storage, or computer services. Each has a unique identifier associated with it for naming purposes.

**Resource Access Control Facility (RACF).** An IBM licensed program, that provides for access control by identifying and verifying the users to the system, authorizing access to protected resources, and logging the detected unauthorized access to protected resources.

**RPC.** Remote Procedure Call.

**RPC interface.** A logical group of operations, data types, and constant declarations that serves as a network contract for a client to request a procedure in a server. See also *interface definition* and *operation*.

**RPC protocol.** An RPC-specific communications protocol that supports the semantics of the DCE RPC API and runs over either connectionless or connection-oriented communications protocols.

**RPC protocol sequence.** A valid combination of communications protocols represented by a character string. Each RPC protocol sequence typically includes three protocols: a network protocol, a transport protocol, and an RPC protocol that works with the network and transport protocols. See *network protocol*, *RPC protocol*, and *transfer protocol*. Synonymous with *protocol sequence*.

## S

**Security Service.** A DCE component that provides trustworthy identification of users, secure communications, and controlled access to resources in a distributed system.

**server.** (1) On a network, the computer that contains programs, data, or provides the facilities that other computers on the network can access. (2) The party that receives remote procedure calls. Contrast with *client*.

**service.** In network architecture, the capabilities that the layers closer to the physical media provide to the layers closer to the end user.

**sign-on.** (1) A procedure to be followed at a terminal or workstation to establish a link to a computer. (2) To begin a session at a workstation. (3) Same as log on or log in.

**single sign-on.** In z/OS DCE, single sign-on to DCE allows a z/OS user who has already been authenticated to an MVS external security manager, such as RACF, to be logged in to DCE. DCE does this automatically when a DCE application is started, if the user is not already logged in to DCE.

**specific.** XOM: The attribute types that can appear in an instance of a given class, but not in an instance of its superclasses.

**standard.** A model that is established and widely used.

**subject identifier (SID).** A string that identifies a user or set of users. Each SID consists of three fields in the form person.group.organization. In an account, each field must have a specific value; in an access control list (ACL) entry, one or more fields may use a wildcard.

**synchronization.** DTS: The process by which a Distributed Time Service entity requests clock values from other systems, computes a new time from the values, and adjusts its system clock to the new time.

## T

**thread.** A single sequential flow of control within a process.

**thread handle.** RPC: A data item that enables threads to share a storage management environment.

**Threads Service.** A DCE component that provides portable facilities that support concurrent programming. The threads service includes operations to create and control multiple threads of execution in a single process

and to synchronize access to global data within an application.

**transaction.** (1) A unit of processing consisting of one or more application programs initiated by a single request, often from a terminal. (2) IMS/ESA: A message destined for an application program.

**transfer syntax.** RPC: A set of encoding rules used for transmitting data over a network and for converting application data to and from different local data representations. See also *Network Data Representation*.

## U

**user.** A person who requires the services of a computing system.

## V

**vendor.** Supplier of software products.

## W

**workstation.** A device that enables users to transmit information to or receive information from a computer, for example, a display station or printer.



---

# Bibliography

This bibliography is a list of publications for z/OS DCE and other products. The complete title, order number, and a brief description is given for each publication.

---

## z/OS DCE Publications

This section lists and provides a brief description of each publication in the z/OS DCE library.

### Overview

- *z/OS DCE Introduction*, GC24-5911

This book introduces z/OS DCE. Whether you are a system manager, technical planner, z/OS system programmer, or application programmer, it will help you understand DCE and evaluate the uses and benefits of including z/OS DCE as part of your information processing environment.

### Planning

- *z/OS DCE Planning*, GC24-5913

This book helps you plan for the organization and installation of z/OS DCE. It discusses the benefits of distributed computing in general and describes how to develop plans for a distributed system in a z/OS environment.

### Administration

- *z/OS DCE Configuring and Getting Started*, SC24-5910

This book helps system and network administrators configure z/OS DCE.

- *z/OS DCE Administration Guide*, SC24-5904

This book helps system and network administrators understand z/OS DCE and tells how to administer it from the batch, TSO, and shell environments.

- *z/OS DCE Command Reference*, SC24-5909

This book provides reference information for the commands that system and network administrators use to work with z/OS DCE.

- *z/OS DCE User's Guide*, SC24-5914

This book describes how to use z/OS DCE to work with your user account, use the directory service,

work with namespaces, and change access to objects that you own.

### Application Development

- *z/OS DCE Application Development Guide: Introduction and Style*, SC24-5907

This book assists you in designing, writing, compiling, linking, and running distributed applications in z/OS DCE.

- *z/OS DCE Application Development Guide: Core Components*, SC24-5905

This book assists programmers in developing applications using application facilities, threads, remote procedure calls, distributed time service, and security service.

- *z/OS DCE Application Development Guide: Directory Services*, SC24-5906

This book describes the z/OS DCE directory service and assists programmers in developing applications for the cell directory service and the global directory service.

- *z/OS DCE Application Development Reference*, SC24-5908

This book explains the DCE Application Program Interfaces (APIs) that you can use to write distributed applications on z/OS DCE.

### Reference

- *z/OS DCE Messages and Codes*, SC24-5912

This book provides detailed explanations and recovery actions for the messages, status codes, and exception codes issued by z/OS DCE.

---

## z/OS SecureWay® Security Server Publications

This section lists and provides a brief description of books in the z/OS SecureWay Security Server library that may be needed for z/OS SecureWay Security Server DCE and for RACF® interoperability.

- *z/OS SecureWay Security Server DCE Overview*, GC24-5921

This book describes the z/OS SecureWay Security Server DCE and provides z/OS SecureWay Security Server DCE information about the z/OS DCE library.

- *z/OS SecureWay Security Server LDAP Client Programming*, SC24-5924

This book describes the Lightweight Directory Access Protocol (LDAP) client APIs that you can use to write distributed applications on z/OS DCE and gives you information on how to develop LDAP applications.

- *z/OS SecureWay Security Server RACF Security Administrator's Guide*, SA22-7683.

This book explains RACF concepts and describes how to plan for and implement RACF.

- *z/OS SecureWay Security Server LDAP Server Administration and Use*, SC24-5923

This book describes how to install, configure, and run the LDAP server. It is intended for administrators who will maintain the server and database.

- *z/OS SecureWay Security Server Firewall Technologies*, SC24-5922

This book provides the configuration, commands, messages, examples and problem determination for the z/OS Firewall Technologies. It is intended for network or system security administrators who install, administer and use the z/OS Firewall Technologies.

## Tool Control Language Publication

- *Tcl and the Tk Toolkit*, John K. Osterhout, (c)1994, Addison—Wesley Publishing Company.

This non-IBM book on the Tool Control Language is useful for application developers, DCECP script writers, and end users.

## IBM C/C++ Language Publication

- *z/OS C/C++ Programming Guide*, SC09-4765

This book describes how to develop applications in the C/C++ language in z/OS.

## z/OS DCE Application Support Publications

This section lists and provides a brief description of each publication in the z/OS DCE Application Support library.

- *z/OS DCE Application Support Configuration and Administration Guide*, SC24-5903

This book helps system and network administrators understand and administer Application Support.

- *z/OS DCE Application Support Programming Guide*, SC24-5902

This book provides information on using Application Support to develop applications that can access CICS® and IMS™ transactions.

---

## Encina Publications

- *z/OS Encina Toolkit Executive Guide and Reference*, SC24-5919

This book discusses writing Encina applications for z/OS.

- *z/OS Encina Transactional RPC Support for IMS*, SC24-5920

This book is to help software designers and programmers extend their IMS transaction applications to participate in a distributed, transactional client/server application.



---

# Index

## A

**Advanced Program-to-Program Communications (APPC)** 14  
**applications, existing cooperative** 14

## B

**bibliography** 37  
**books, list of DCE and related** 37

## C

**cells** 16  
**CICS access** 19  
**client server model** 7  
**components, DCE** 11

## D

**DCE AS CICS** 19  
**DCE AS IMS** 19  
**Directory** 10  
**distributed computing**  
    **benefits** 2  
    **client/server** 7  
    **description** 1  
    **need** 1  
    **solution** 2  
**Distributed Computing Environment (DCE)**  
    **architecture** 9  
    **goals** 7  
    **services**  
        **Directory** 10  
        **RPC** 10  
        **Security** 10  
        **Threads** 10  
        **Time** 10  
**Distributed Time Service** 10

## G

**glossary** 31

## I

**IMS access** 19  
**integration, service** 11

## O

**open systems**  
    **description** 3  
    **IBM** 4

**open systems** (*continued*)

**OSF** 3

**OSF (Open Software Foundation)** 3

## R

**Remote Procedure Call** 10

## S

**scenarios**  
    **education** 25  
    **manufacturing** 23  
**Security** 10  
**service integration** 11

## T

**Threads** 10

## U

**user views** 21

## Z

**z/OS**  
    **description** 4  
    **in DCE** 13  
**z/OS DCE**  
    **architecture** 15  
    **CICS access** 19  
    **IMS access** 19  
    **strengths** 13



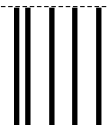




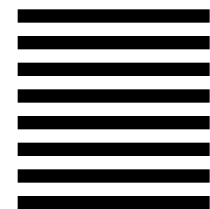
Fold and Tape

Please do not staple

Fold and Tape



NO POSTAGE  
NECESSARY  
IF MAILED IN THE  
UNITED STATES



# BUSINESS REPLY MAIL

FIRST-CLASS MAIL PERMIT NO. 40 ARMONK, NEW YORK

POSTAGE WILL BE PAID BY ADDRESSEE

Department G60  
International Business Machines Corporation  
Information Development  
1701 North Street  
ENDICOTT NY 13760-5553



Fold and Tape

Please do not staple

Fold and Tape







Program Number: 5694-A01



Printed in the United States of America  
on recycled paper containing 10%  
recovered post-consumer fiber.

GC24-5911-00





**z/OS DCE**

**Introduction**