SG24-4752-00

Addressing OLTP Solutions with CICS: The Transaction Server for AIX

September 1996

International Technical Support Organization
San Jose Center

IBM

International Technical Support Organization

SG24-4752-00

Addressing OLTP Solutions with CICS: The Transaction Server for AIX

September 1996

Take Note!

Before using this information and the product it supports, be sure to read the general information in Appendix A, “Special Notices” on page 171.

First Edition (September 1996)

This edition applies to IBM Transaction Server for AIX Version 4, Program number 5697-251 for use with the AIX Operating System Version 4 Release 1 Modification 4 or above.

Comments may be addressed to:
IBM Corporation, International Technical Support Organization
Dept. 471 Building 80-E2
650 Harry Road
San Jose, California 95120-6099

Contents

Figures

Tables

Preface

This book is unique in its detailed coverage of the IBM Transaction Server for AIX with CICS for AIX as the transaction monitor. It focuses on CICS for AIX as the core of a comprehensive online transaction processing (OLTP) solution and how it interacts with other components, such as CICS Clients and CICS System Manager for AIX. The emphasis is on planning considerations. The book provides information about the functions and services supplied by the components of the OLTP solution and explains what you should consider during the planning phase of implementing of the solution.

This book is written to help you understand how CICS for AIX and the associated software products supplied in the IBM Transaction Server for AIX work together to provide you with the most comprehensive, integrated, customized solutions for your complex OLTP business environment. Some basic knowledge of OLTP is assumed.

This book is suitable for:

Customers To help them visualize the enormous potential that the IBM Transaction Server for AIX and CICS for AIX have to offer as a comprehensive OLTP solution

Sales representatives To help them understand the flexibility of the IBM Transaction Server for AIX and CICS for AIX OLTP solution

Systems engineers To help them plan for the implementation of the IBM Transaction Server for AIX and CICS for AIX OLTP solution

(187 pages)

How This Redbook Is Organized

This book is organized as follows:

- Chapter 1, "Introduction"

  This chapter describes the complex OLTP environment that enterprises face.

- Chapter 2, "IBM Transaction Server Overview"

  This chapter provides an overview of IBM Transaction Server for AIX and its relationship with other IBM Software Servers.

- Chapter 3, "CICS for AIX"

  This chapter describes the services provided by CICS for AIX as the transaction monitor. Planning considerations for its implementation are presented.

- Chapter 4, "Using CICS Clients"

  This chapter provides an overview of the CICS Clients and the planning considerations for a variety of CICS clients supplied with the IBM Transaction Server for AIX.

- Chapter 5, "Interactions with DCE"

This chapter describes the infrastructure provided by DCE to support the proper execution of a CICS for AIX system. Planning considerations are discussed.

- Chapter 6, "Interactions with Encina for AIX"

  This chapter describes the transactional services provided by Encina to support the CICS for AIX system. Planning considerations are discussed.

- Chapter 7, "Interactions with the DB2 Family of Products"

  This chapter provides an overview of the X/Open XA interface. Various RDBMS access options are discussed with reference to DB2 as the RDBMS. Planning considerations for accessing various databases belonging to the DB2 family from the CICS for AIX system are presented.

- Chapter 8, "Using CICS System Manager for AIX"

  This chapter describes the functions and components of CICS System Manager for AIX and explains how it can be used to manage the CICS for AIX systems in terms of workload and system management. Planning considerations are presented.

- Chapter 9, "Accessing CICS from the Web"

  This chapter describes how CICS applications running in the CICS for AIX system can be accessed from the World Wide Web using CICS Internet Gateway for AIX supplied by IBM Transaction Server for AIX. Relevant planning considerations are discussed.

- Chapter 10, "Application Design Considerations"

  This chapter provides the considerations for designing good CICS applications in general.

- Chapter 11, "TDJ Enterprise International Limited"

  This chapter describes a sample complex OLTP environment to demonstrate how IBM Transaction Server for AIX has helped facilitate a complex OLTP.

The Team That Wrote This Redbook

This book was produced by a team of specialists from around the world working at the International Technical Support Organization, San Jose Center.

Guido De Simoni is a Project Leader at the International Technical Support Organization, San Jose Center. He writes extensively and teaches IBM classes worldwide on all areas of distributed OLTP. Before joining the ITSO three years ago, Guido worked in the AIX Competency Center in Italy as a Connectivity Advisor.

Tim Dunn is a Performance Analyst at the Hursley Laboratory in England. He has eight years of experience in the performance tuning of CICS for AIX and CICS for MVS/ESA systems. He has also worked extensively with CICS System Manager for AIX, focusing on the workload management component.

Haiyan Jia is a Systems Engineer at IBM China Co. He has two years of experience in technical support of CICS for AIX systems. Haiyan also worked as a UNIX systems administrator for three years.

Yan Tsang is an Advisory IP Specialist at the IBM Support Center in Sydney, Australia. He has 21 years of experience in various support services and for the

CICS family of products. Yan helped in the design of the CICS Automated Operation product for the MVS environment. His recent service activities have focused on solutions using CICS and MQSeries on AIX systems.

Thanks to the following people who took time to review this book:

Maurizio Canton
IBM Italy

Barbara Ioris
IBM US Transaction Systems Marketing

Silvio Podcameni
International Technical Support Organization, San Jose Center

Geoff Sharman
IBM Hursley Laboratories

Thanks also to Maggie Cutler for her editorial review and Christine Dorr for her graphics support.

Comments Welcome

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

redbook@vnet.ibm.com

Your comments are important to us!

Part 1. The Problem

Online transaction processing (OLTP) systems are becoming an increasingly common part of our daily lives, often without our realizing it. Whether you are withdrawing cash from an automatic teller machine or using your credit card to buy gas or groceries, you are interacting with OLTP systems. Because of the problems that they must solve, OLTP systems are necessarily complex.

In Part 1 of the book we discuss the nature of complex OLTP systems. We also look at the requirements that must be met to keep current systems running and enable them to expand as new business opportunities arise.

Chapter 1. Introduction

During the past three decades, online transaction processing (OLTP) has evolved from the simple network of dumb terminals connected to a host computer that runs simple inquiry business transaction applications to a sophisticated, heterogeneous network of intellegent workstations and personal computers (PCs) acting as clients and connecting to clusters of server computers that run mission-critical transaction applications. OLTP systems are typically the lifeblood of an enterprise′s computer systems. They are critical to the fulfillment of the mission of the enterprise, whether the enterprise is a commercial company, a government institution, or a community organization. Millions of transactions and billions of dollars are processed everyday with systems running OLTP applications. Today OLTP is the foundation of data processing throughout the business world.

In this chapter we cover the common properties of an OLTP system and the driving forces creating the complex OLTP business environment. We explain the meaning of an OLTP complex, discuss distributed OLTP, and look at the OLTP marketplace.

1.1 What Is OLTP?

In simple terms OLTP refers to business transactions that are processed immediately on computer systems. Examples from daily life are automated teller banking and point of sales applications in retail shops. For a system to be classified as an OLTP system, it must exhibit the following four ACID properties:

A	Atomicity
C	Consistency
I	Isolation
D	Durability

The sequence of actions taken by a transaction must be **atomic**, that is, indivisible. Either the whole sequence completes or the whole sequence does not occur at all. Atomicity ensures that no sequence of actions partially executes and data integrity is maintained.

The changes to the data made by a transaction must be **consistent**. Enterprise data represents the state of the enterprise, and any change must keep all of the data consistent.

A transaction appears to run by itself, and other transactions appear to run either before it or after it. Such **isolation** is required to enable concurrent execution of transactions.

The changes made by a completed transaction must survive failures of any kind and be permanent. This property is called **durability**.

For more information about ACID properties, please see J. Gray and A. Reuter, *Transaction Processing: Concepts and Techniques*. San Mateo, CA: Morgan Kaufman Publishers, 1993.

1.2 Complex OLTP Business Environment

Intensified competition, globalization of international trade, rapidly changing markets, a changing workforce with greater accountability, and evolving information technology (IT) create enormous pressure on most enterprises. To stay competitive and be prosperous, an enterprise must go through major transformations in the form of:

- Reengineering existing business processes
- Redeploying the workforce effectively
- Restructuring the business organization with fewer levels of management
- Redesigning every phase of the business with enough flexibility to build new products and services for rapidly changing markets

Instead of the hierarchical structure in the past, the enterprise takes on a network structure. Internally this means that people from different disciplines and different geographical locations make up the project team. Externally, customers, business partners, and vendors form the dynamic network structure, which exists as long as the parties involved see value in maintaining such structure.

Multiple small markets replace the huge monolithic market to meet the specific requirements of assertive and informed customers. With the advancements in IT to provide affordable computing power at the desktop, enterprise workers on the front line of customer contact demand the ability to act on current information and make informed decisions on the spot to serve their customers well. They become the knowledge workers who bear responsibility for boosting customer satisfaction and therefore the profitability of the enterprise.

These multiple transformations come together to create a complex business environment for OLTP (Figure 1 on page 5). The customers (end users) of OLTP applications now come from within and outside the enterprise. They may reside in different geographic locations with different types of computers on their desktop. These give rise to potentially complex networks over which OLTP applications have to execute.

In addition enterprise data is no longer centralized at the host computer. Most likely it is distributed across the enterprise in the form of different types of databases residing on a variety of different computer platforms. The variety of operating systems running in these different platforms adds more complexity to an already complex environment.

Furthermore, as more and more new IT products and solutions hit the market at an ever-increasing rate, their adoption and integration impose another dimension of complexity.

The ultimate objective of an OLTP system is to execute transactions to meet increasing business demand. The transactions themselves are becoming more complex in nature because they reflect the business reality of the enterprise. Therefore there is a great demand for comprehensive solutions to overcome the complexity that comes with complex OLTP business environment.

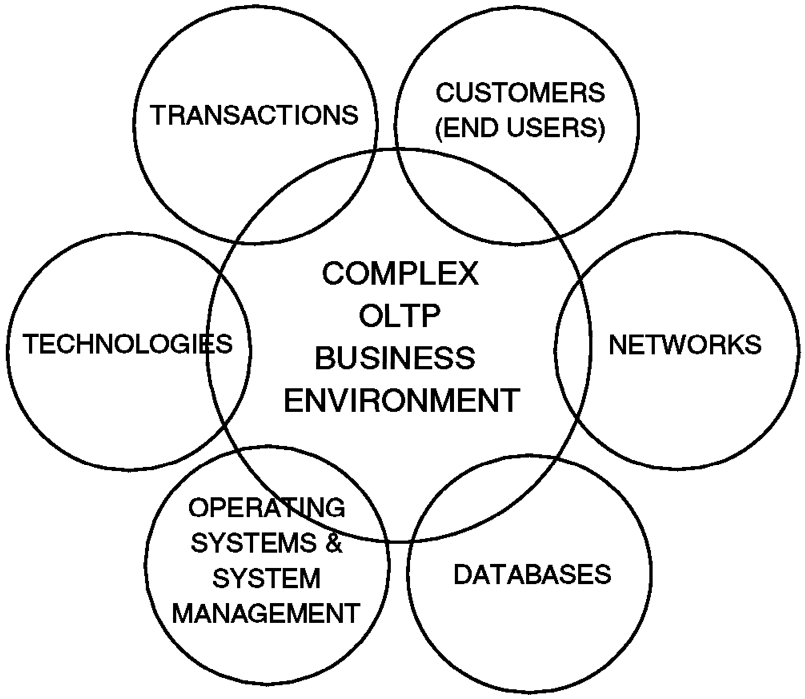

Figure 1. Components of Complex OLTP Business Environment

1.3 OLTP Complex

Sophisticated transaction processing capabilities are required more than ever to meet the comprehensive needs of OLTP application management, development and deployment; thus the need for transaction processing monitors (TP monitors) and application servers. The OLTP complex in which the enterprise runs its mission-critical OLTP applications represents the physical view of the environment that supports the execution of the OLTP applications. The OLTP complex involves a number of discipline areas such as the operating systems, database, transaction processing, security, networking, system management, application systems, and distributed systems infrastructure.

We illustrate the OLTP complex with two configurations. Figure 2 on page 6 shows an OLTP complex for the XYZ Insurance Company. Basically the design follows the so-called application server model, which consists of database servers, business application servers, and the terminal servers providing presentation services to the end users. (For more detail about the application server model, please see 4.1, "Application Server Model" on page 49.) The company, which moved its OLTP applications from the legacy host system to the open distributed platform, has an OLTP complex from every point of view. Instead of having one monolithic central host computer, it now has nine server computers with a multitude of software for administration.

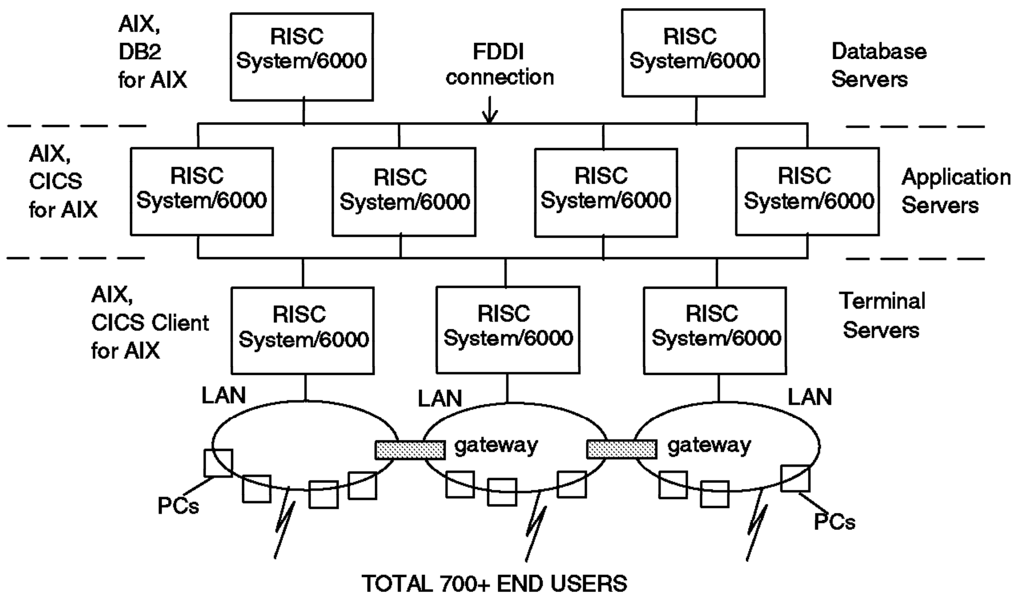

Figure 2. OLTP Complex for XYZ Insurance Company

Figure 3 shows an OLTP complex for the ABC Banking Corporation, which has to provide telebanking facilities to its retail customers to stay competitive. By linking the four server computers to the host computers that act as distributed servers, the telebanking OLTP applications can access data at the host computers. Again it is an OLTP complex.

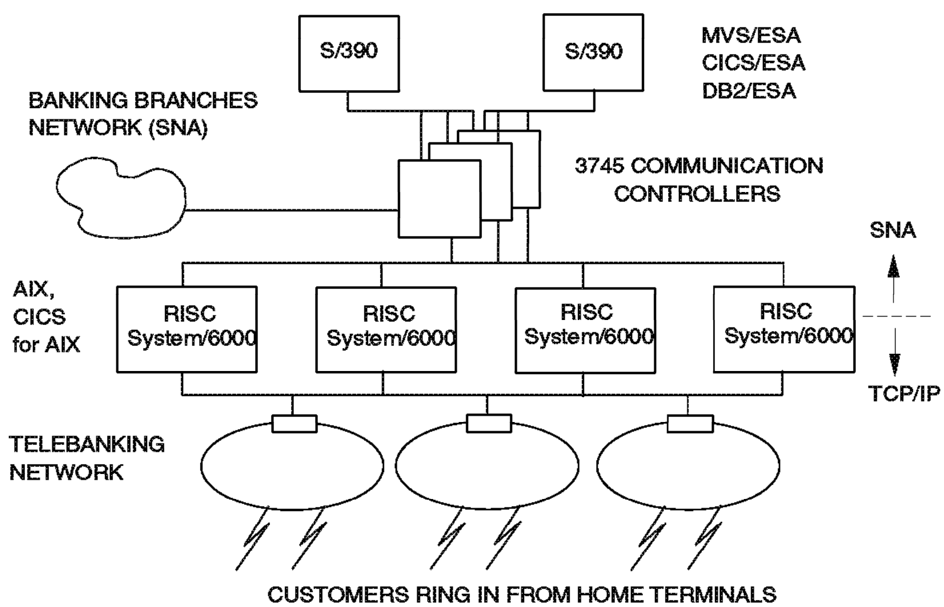

Figure 3. OLTP Complex for ABC Banking Corporation

Both enterprises implement an OLTP complex to answer their requirements:

- High availability

  Mission-critical OLTP applications require absolute data integrity and no downtime, either planned or unplanned. The uptime of the OLTP system should be maximized to meet business needs.

- High reliability

Each component in the complex OLTP environment has to be highly reliable so that the resultant system becomes highly reliable, with minimum downtime and minimum repair time.

- Recoverability

  The ability to recover lost data quickly from any failure is very important. As failures occur, the system, including databases, must be rapidly restored to the consistent point before the failure.

- Scalability

  The system is regarded as scalable if it can be expanded in increments with a highly flexible set of upgrades at minimum cost.

- Operability

  For highly distributed, complex OLTP environments, appropriate tools must be available to facilitate network, system, and user administration.

- Productivity

  Sophisticated application development tools are required to develop and maintain OLTP applications. These tools should help existing applications evolve toward new technologies or models, for example, object-oriented solutions.

- Connectivity

  All commonly used network connections, such as LAN, WAN, SNA, TCP/IP, and PC connectivity, should be supported. Data access between different networks should be supported in addition to terminal access and file transfer.

- Distributability

  As OLTP environments become and more more complex and distributed, transactions must be managed across multiple sites and multiple, heterogeneous systems and databases.

- Data integrity

  The consistent state of the enterprise data must be maintained. Facilities such as comprehensive error recovery, rigorous application design, and transactional integrity support ensure that no lost transactions occur and the durability and completeness of business transactions are preserved.

- Security

  Enterprise data is a valuable asset. Its access by users and applications must be secure in the increasingly distributed open system environments.

- Price-to-performance

  The industry standard benchmarks for OLTP are the Transaction Processing Council (TPC) benchmarks, which include the TPC-C and TPC-D benchmarks. The metrics reported include throughput expressed in transactions per second (TPS) and the associated costs ($)/TPS. The systems must perform well and be reasonably priced.

- Openness

  The degree of openness of a system depends on its scalability, portability, and interoperability.

- Portability

By using common application programming interfaces (APIs), applications developed on one platform can run on others.

- Interoperability

  Distributed applications can work together with the common communication protocols.

- Evolution

  Taking on new techologies and new standards for protocols and interfaces becomes easy with modular designs and well-specified internal interfaces of the truly open system.

- Investment protection

  Existing systems and staff skills are the defense against failure of mission-critical OLTP applications on which the enterprise depends. Therefore the current investments in those systems and skills must be protected and preserved. In other words, there is no obsolescence, only enhancement through the use of open distributed OLTP systems to operate with the existing proprietary computing platforms.

- Overall cost reductions

  Through the focus on improving return on investment, making more productive use of ITs, and careful selection of appropriate platforms for the OLTP applications, the gradual planned shift to open distributed systems will generate overall cost reductions.

- Ease of reconfiguration

  The system has to be easily reconfigurable to match the ever-changing needs of the business.

- Workload management

  The ability to distribute and balance the workload among all transaction processing servers is becoming important because it enables the delivery of consistent service levels to end users.

- Multivendor support

  Heterogeneous multivendor systems and networks are becoming the norm. Support for multivendor environments is a must for any serious players in the OLTP market.

- Coping with business changes

  Business changes demand corresponding and timely adjustments in the system.

- Computing resource sharing

  Some of the computing resources are still expensive; sharing them among applications and end users makes business sense.

- Data sharing

  Data sharing enables different applications from distributed sites to use the same data. It eliminates the duplication of data that creates complexity in data management.

In Table 1 on page 9 we tabulate the above requirements with respect to the specific disciplines to show how well these disciplines meet the requirements.

Table 1. How Well Specific Disciplines of the OLTP Complex Meet Enterprise Requirements

User Requirements	Operating System	Database	Online Transaction Processing	Security	Network	Systems Management	Applications	Distributed Systems Tools
High availability	YES	YES	YES	YES	YES	YES	YES	YES
High reliability	YES	YES	YES	YES	YES	YES	YES	YES
Recoverability	YES	YES	YES	YES	YES	YES	YES	YES
Scalability	YES	YES	YES		YES	YES		
Operability	YES	YES	YES	YES	YES	YES	YES	YES
Productivity	YES	YES	YES			YES		
Connectivity		YES	YES		YES	YES		YES
Distributability	YES	YES	YES		YES	YES	YES	YES
Data integrity	YES	YES	YES	YES		YES	YES	
Security	YES	YES	YES	YES	YES	YES	YES	YES
Price-to-performance	YES	YES	YES	YES	YES	YES	YES	YES
Openness	YES	YES	YES		YES	YES		
Portability	YES	YES	YES			YES	YES	
Interoperability	YES	YES	YES		YES	YES		
Evolution	YES	YES	YES	YES	YES	YES		
Investment protection	YES	YES	YES		YES			
Overall cost reduction	YES	YES	YES			YES		
Ease of reconfiguration	YES		YES		YES	YES		
Workload management			YES			YES		
Multivendors support		YES	YES		YES			YES
Coping with business changes	YES	YES	YES		YES	YES		
Computing resource sharing	YES	YES	YES		YES	YES		
Data sharing		YES	YES					

Key:

YES = the specific user requirement is well met.
blank = the specific user requirement is not well met.

1.4 Distributed OLTP

Many enterprises prefer to develop new OLTP systems based on a distributed processing model. The advantages of using less expensive, more responsive, more user friendly new systems in a networked environment outweigh the pitfalls created by the complexity of the configuration. Instead of the classic model of a highly tuned, very powerful, central host system providing the total OLTP capability, the distributed OLTP model is made up of a hierarchy of systems, from desktop PCs through mid-range server computers to large-scale server computers. The end-user input devices can range from character-based terminals to very powerful workstations. With multiple layers of servers capable of executing transactions, the incoming transaction request can be routed over the network to the appropriate server computer for execution. The turnaround time of running the same transaction can vary greatly according to the location of the data accessed by the transaction.

Figure 4 on page 10 shows a distributed OLTP system with a hierarchy of servers. In each server, the TP monitor provides the routing function of the transaction. If the TP monitor evaluates and decides to route the transaction request, it makes sure that the transactions definitely arrive at their destinations and the results are returned correctly over the potentially complex network. Furthermore the TP monitor helps to optimize the performance of the server computer and enhances reliability.

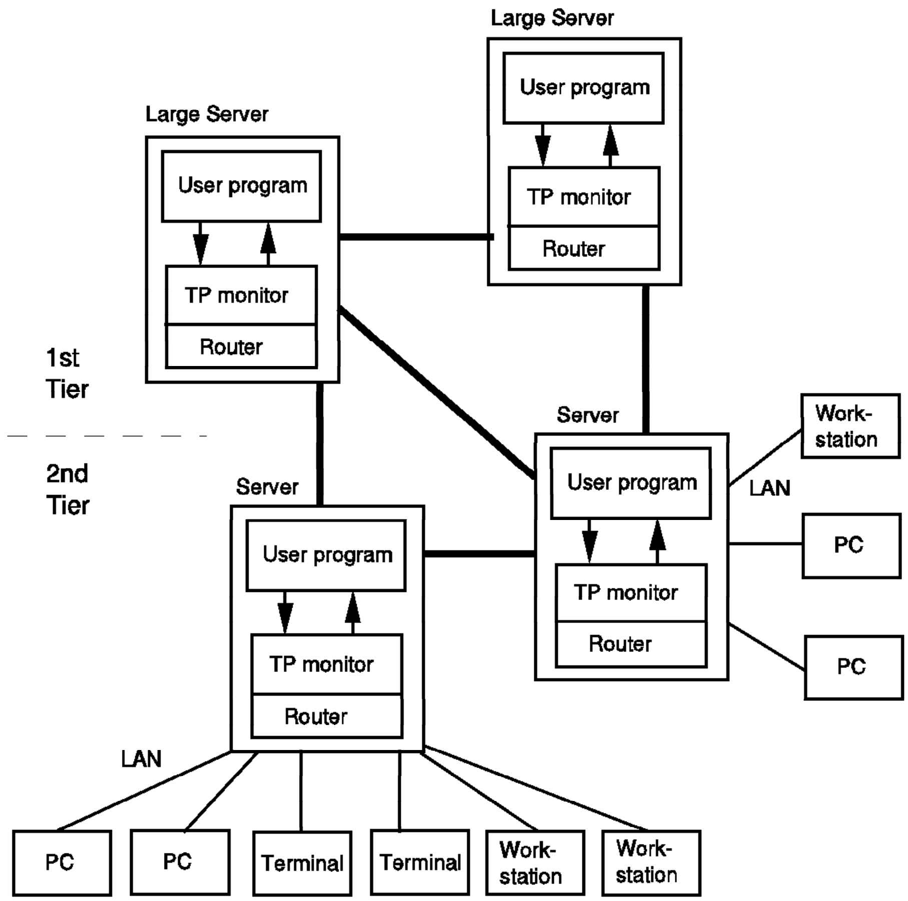

Figure 4. A Distributed OLTP System

At this point we expect you may be a bit confused about the OLTP complex and distributed OLTP. You may think that they are the same. We would like to clarify this. Distributed OLTP is a subset of the OLTP complex. In the simplest case, the OLTP complex consists of one distributed OLTP; then they are the same. Within an enterprise, a number of discrete OLTP systems that are distributed in functions and data can exist along the boundaries of the business units they serve. In the end these separate systems have to operate as one total system for the whole enterprise. We refer to this total system as the *OLTP complex*.

1.5 OLTP Market

The OLTP market is huge in both financial and technological terms. No serious players in the IT industry would ignore this market. To analyze this market in a logical way, we can broadly categorize the enterprises using OLTP into four quadrants based on a segmentation model (see Figure 5 on page 12). The OLTP market can be segmented according to the motives behind the deployment of the OLTP system by the enterprise.

In the first quadrant, the enterprises mainly move their existing OLTP applications from their current host systems to the alternative platforms as a way of reducing the operational costs and the first step toward the readiness for a highly distributed, open, OLTP system configuration. The enterprises can be regarded as straightforward downsizers.

In the second quadrant, the enterprises purchase the readymade OLTP application solutions relevant to their industries as the means of assisting and supporting their business reengineering processes. As these are new OLTP applications most likely ready for a distributed open environment, it would be appropriate to implement them on the alternative server platforms. It is a direct replacement approach. The new configurations replace the existing OLTP applications and the associated host systems.

In the third quadrant, the enterprises implement the three-tier client/server OLTP applications both to access host data and to interconnect existing OLTP applications running on different platforms. The three-tier client/server model is the same concept as the application server model. The resultant configuration is the distributed OLTP system.

In the fourth quadrant, the enterprises target the ultimate goal of implementing enterprisewide client/server OLTP systems. The current host systems participate as distributed servers in the enterprisewide distributed networks interconnecting multivendor server computers and client computers. All of these computers work together to provide the full-function, enterprisewide OLTP system serving all eligible end users of the enterprise.

As you can see in Figure 5 on page 12, the specific needs of the enterprise in each quadrant are quite different. Therefore the comprehensive OLTP solutions provided by the vendors must be highly flexible and meet most, if not all, of the enterprise requirements listed in Table 1 on page 9.

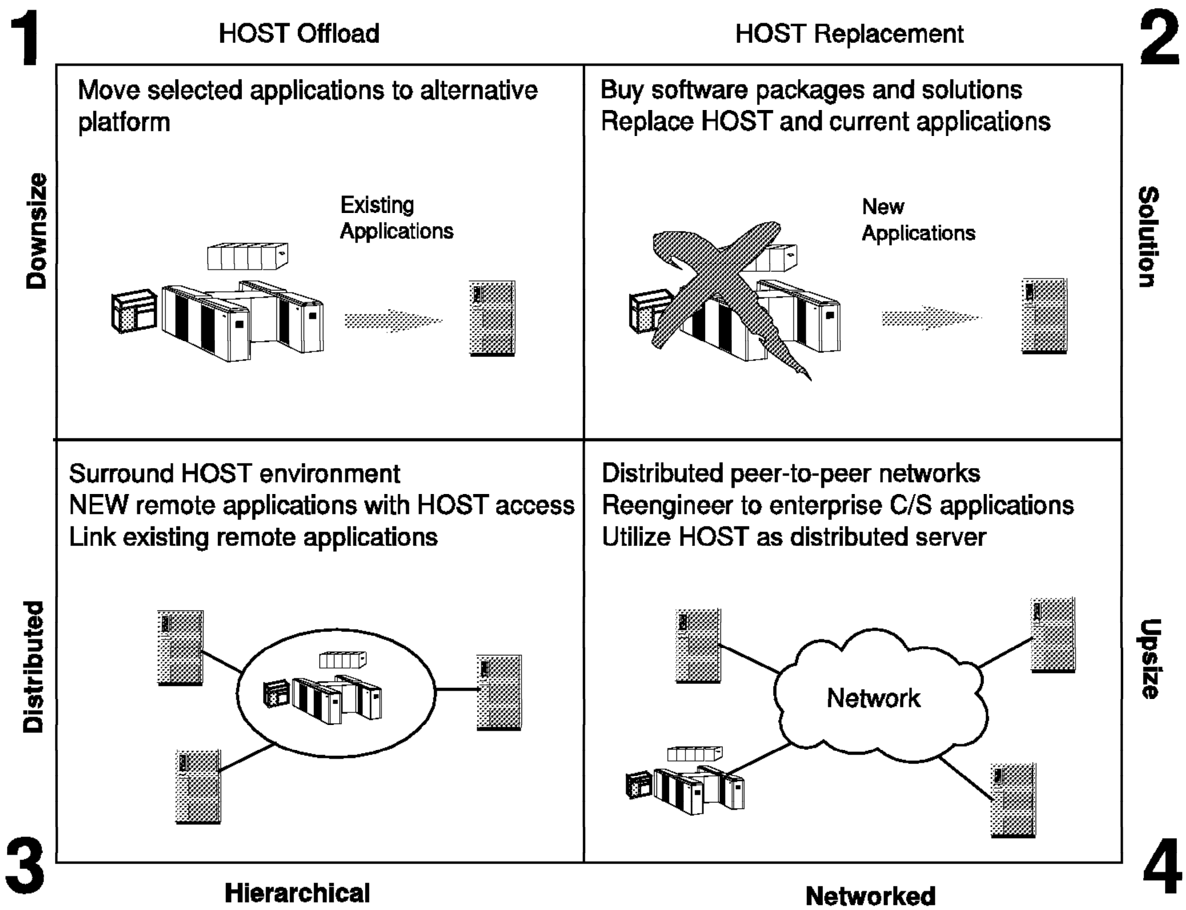

Figure 5. OLTP Market Segmentation Model

1.6 Summary

The big challenge that enterprises face in the area of OLTP is complexity. Good, comprehensive solutions are in great demand. In Part 2 of this book we describe the best solution available: **IBM Transaction Server for AIX Version 4** and **CICS for AIX**.

Part 2. The Solution

Today's OLTP systems are so varied and complex that is beyond the scope of one software component or product to solve all of needs an enterprise's requirements for implementing its business systems. The development of complete, robust, and managed systems can occur only by combining software components into packages.

In Part 2 of this book we show how the IBM Transaction Server for AIX and CICS for AIX can be used as the foundation on which you can build your distributed enterprise business systems.

Taking CICS for AIX as the core enabler of our OLTP system, we examine how other components in the IBM Transaction Server for AIX and other IBM Software Servers can be combined to provide a framework for highly interconnectable, robust, and reliable business systems. For each of these other components we examine the benefits they bring to the total environment and how they can be combined with CICS for AIX. We then discuss the planning issues associated with combining these components with CICS for AIX. We aim to show the high degree of flexibility that exists within a CICS for AIX based environment. Whether you want to access existing enterprise data or make the best use of existing Intel-based hardware, for example, CICS for AIX and other members of the CICS family provide the right facilities.

Chapter 2. IBM Transaction Server Overview

In March 1996, IBM announced a family of IBM Software Servers that would enable customers to rapidly implement client/server applications as well as extend applications to meet future business requirements. The following servers were announced:

- Lotus Notes
- IBM Communications Server
- IBM Directory and Security Server
- IBM Database Server
- IBM SystemView Server
- IBM Internet Connection Server
- IBM Transaction Server

The aim of these software servers is to meet today′s need for software that is easy to install, use, and operate while working with existing systems. Each software area covers a specific functional area. In this book, we focus on complex OLTP environments. The software server we focus on is therefore the IBM Transaction Server.

An enterprise has many issues to consider when implementing IT systems. The requirements extend beyond transaction processing to areas such as office automation, database management, system and network management, connectivity, and security management. Thus it might be necessary to use more than a single software server to adequately address the needs of a particular application (see Figure 6).

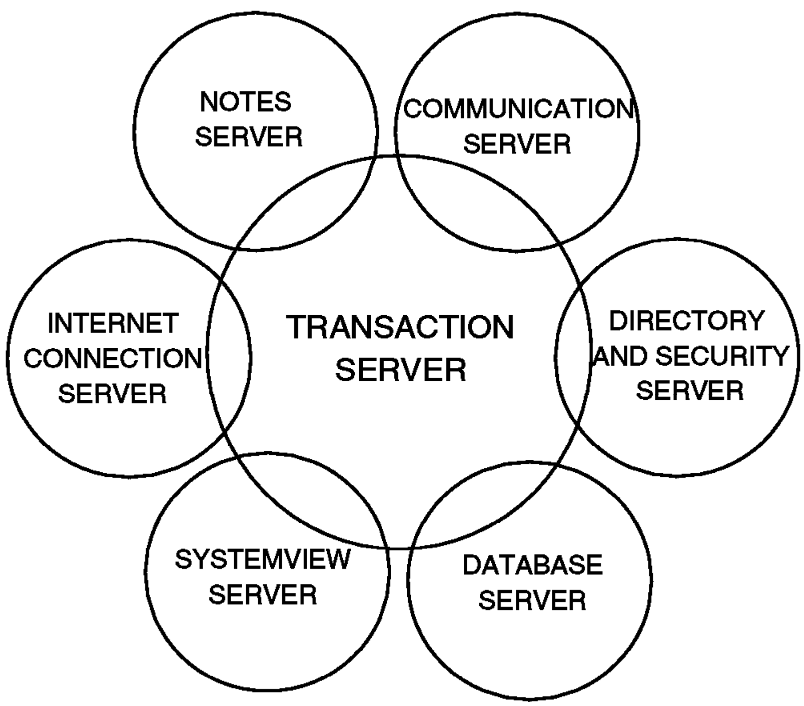

Figure 6. IBM Software Servers

2.1 IBM Software Servers

In this section we provide an overview of all of the IBM Software Servers except the IBM Transaction Server, which we covered in detail in 2.2, “IBM Transaction Server” on page 17.

2.1.1 Lotus Notes

Lotus Notes Release 4 allows you to find, manage, and share all information you need. Notes has a new, intuitive user interface that is based on the market-leading product cc:Mail. Whether it is on the Internet, in relational databases or host systems, in a desktop application, or in E-mail, Lotus Notes provides an easy-to-use central access point for data sharing.

With the Notes new Release 4 InterNotes Web Navigator, you can leverage (not just surf) the Web, capturing and managing relevant business information and putting it to immediate use every day.

Everyone can communicate and collaborate, as individuals or teams, whether they are using Windows, OS/2, Mac, UNIX, or NT. In the office or on the road, ideas and information are continually exchanged and updated, giving you the competitive edge.

With scalable servers supporting up to 1000 users and powerful administration capabilities, managing a Notes system is easy. Thus you can implement custom applications to leverage your greatest asset of all: the combined experience and knowledge of your people.

2.1.2 IBM Communications Server

Systems Network Architecure (SNA) remains a key networking protocol in enterprise environments that include MVS and VM mainframes as well as AS/400 systems. The Communications Server provides SNA and Advanced Peer-to-Peer Networking (APPN) communication for the other IBM Software Servers and clients they support. It also provides transport independence, allowing LU 6.2-based and Sockets-based applications to run over either SNA or TCP/IP networks.

2.1.3 IBM Directory and Security Server

The distributed directory and security functions of the Directory and Security Server are essential to providing transparent access to distributed resources. The Directory and Security Server is based on open technology, namely, the Distributed Computing Environment (DCE) for the Open Software Foundation (OSF), which offers open services for information access in large, heterogeneous networks. The Directory and Security Server is available for the IBM OS/2 Warp and IBM AIX platforms.

2.1.4 IBM Database Server

The Database Server provides management of and access to data stored locally or in remote systems. Access to data through the World Wide Web (WWW) is provided. This full-functioned structured query language (SQL) relational database system is designed to manage data for the individual, the workgroup, and companywide solutions needed in today′s business world.

The Database Server is available on the IBM OS/2 Warp, IBM AIX, and Microsoft Windows NT operating systems. It offers reliable, responsive, and proven features that are implemented using open standards. It also offers trend-setting data storage and access for large text, image, audio, and video objects, plus user-defined data types and functions. The scalability of the Database Server spans uniprocessors, symmetric multiprocessors (SMPs), and massively parallel systems.

The Database Server includes application development software kits to enable client/server systems for a wide range of clients.

2.1.5 IBM SystemView Server

Providing the tools necessary to manage small to medium workgroups as well as growing enterprises, the SystemView Server enables you to manage your entire network and the systems that share the network.

SystemView Server is the first systems management server offering of the IBM Software Servers Family. You can expect to see further developments in this area. IBM has stated its intention to announce enhanced systems management offerings as members of the IBM Software Servers family. These offerings will take advantage of IBM′s enterprise management strengths along with Tivoli Systems′ leading-edge distributed management technologies.

2.1.6 IBM Internet Connection Server

The Internet Connection Server allows a business to establish a presence on the Internet by providing the means to publish company and product information on the WWW. With the increase in popularity of the Internet and the possibilities it holds for business and commerce, the Internet Connection Server should become one of the most popular servers. The Internet Connection Server is a key component of IBM′s network computing strategy.

2.2 IBM Transaction Server

The IBM Transaction Server acts as a robust, enterprisewide coordinator and integrator of servers, managing high-performance applications and data sources across the network. With the Transaction Server, you can create a distributed client/server environment with all of the reliability, availability, and data integrity required for today′s OLTP and business-critical applications. The Transaction Server is based on the proven, robust technology of CICS and Encina, industry-leading transaction processing products.

The Transaction Server, in cooperation with other IBM software products (see Figure 7 on page 18) provides an easily managed client/server infrastructure for developing and deploying business-critical applications both within and beyond organizational boundaries. Thus you have a robust and open platform on which to build your enterprise applications. Because there is choice and flexibility of platforms, components, and programming languages, your applications can be:

- **Highly performing** with a consistent and fast response
- **Portable** across many IBM and non-IBM systems
- **Interoperable** across local area networks (LANs) and wide area networks (WANs), maintaining integrity and dependability
- **Scalable** from tens to thousands of users
- **Manageable** from a single point of control, even in complex configurations

A Transaction Server business application encompasses all of its associated services and resources. Therefore programs, databases, message queues, connections, terminals, security, and users can be administered separately for each application. The Transaction Server coordinates each user transaction as it runs, to ensure the integrity of resources—and of your business operations.

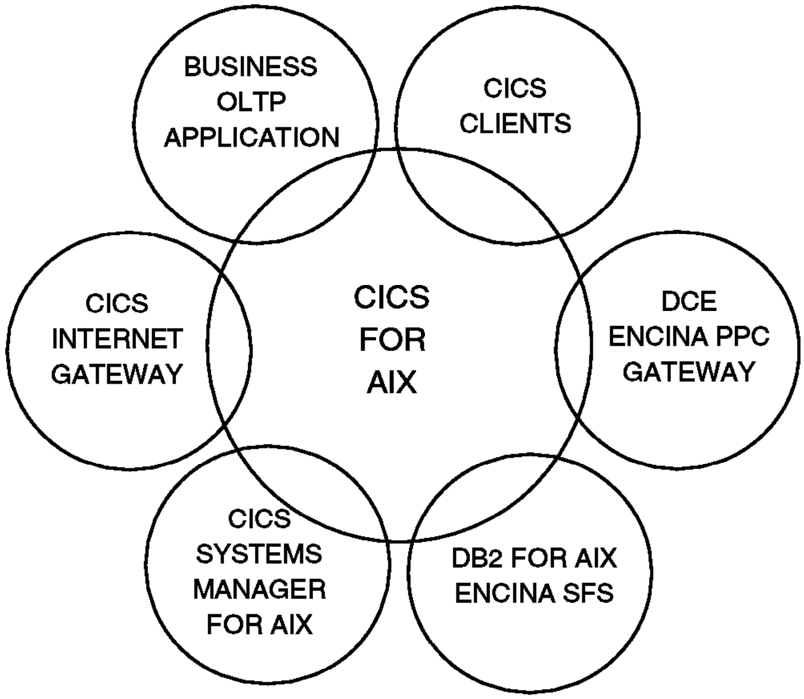

Figure 7. Interaction by IBM Transaction Server with CICS Choice

IBM Transaction Server for AIX version 4 contains:

- CICS Clients, 1.1.0
- Encina for AIX, 2.2
- CICS Systems Manager for AIX, 1.1
- CICS Internet Gateway for AIX, V1.1.0
- CICS for AIX, 2.1.1

2.2.1 CICS Clients V1.1.0

The IBM CICS Clients are a family of client products for workstations running a range of operating systems. The CICS Clients family communicates through a variety of protocols and a range of supported communication products directly with CICS servers on all platforms. The CICS Clients bring the benefits of client/server operation to the entire CICS family of application servers.

The IBM CICS Clients offer clients for a variety of IBM and non-IBM platforms: OS/2, DOS, Windows, Macintosh, AIX, and Sun in addition to the Encina Client for AIX. These clients are capable of communicating over Advanced Program-to-Program Communication (APPC), TCP/IP, and NetBIOS protocols.

IBM CICS Clients can work with CICS servers on ESA, MVS, VSE, AS/400, AIX, and OS/2 platforms.

2.2.2 Encina for AIX V2.2

IBM′s Encina for AIX is a family of products that extends and simplifies the use of the OSF DCE for distributed, transactional client/server, and OLTP computing. Encina provides the DCE-based infrastructure to *glue* together applications distributed across the PC Desktop, UNIX servers, and the mainframe, with complete transactional integrity.

The Encina Monitor Suite for AIX is a single, comprehensive TP monitor server environment for large-scale client/server computing. It is a newly created server package that includes all standard Encina services, thereby enabling customers to more easily purchase and deploy the services across their server machines.

The question of whether to choose CICS for AIX or the Encina Monitor Suite for the development of client/server distributed applications to a large extent depends on your current configuration. For those enterprises that have an existing investment in CICS skills and a mainframe environment, CICS for AIX is recommended. It will enable connectivity to existing CICS regions and hence the data that those CICS systems can access. For those enterprises that have an existing investment in UNIX and C or DCE programming skills, the Encina Monitor Suite is recommended.

The Encina Structured File Server (SFS) is a record-oriented file system that provides full transactional integrity, high performance, and log-based recovery for fast restarts. It is highly scalable to support very large databases and is fully capable of participating in two-phase commit protocols, allowing multiple SFS servers to be used in a single transaction. SFS supports B-tree clustered, relative, and entry-sequence access methods and provides both X/Open ISAM-compliant and VSAM-like interfaces. For increased availability, SFS supports online backup and restore. Administration and access-control-based security are consistent with the services provided by the Encina Monitor and DCE.

The Encina Peer-to-Peer Communication (PPC) Gateway for AIX provides the bridge from TCP/IP to SNA environments. With the Gateway′s SNA Sync Level 2 support, database updates can occur with integrity among applications running on mainframe CICS/ESA systems and Encina for AIX or CICS for AIX systems. Transactional access to the mainframe through the Encina PPC Gateway and SNA Sync Level 2 is a capability unique in the industry and thus provides significant competitive advantage.

2.2.3 CICS Systems Manager for AIX V1.1.0

CICS Systems Manager (CICS SM) provides systems and workload management for CICS for AIX Version 2.1.1 in the SMP and parallel SP AIX environments. CICS System Manager gives you a single point of control and a single system image. It also helps to improve system availability through workload management and brings the power of object-oriented technology to the world of CICS and customer business solutions.

2.2.4 CICS Internet Gateway for AIX V1.1.0

The CICS Internet Gateway enables you to access and run many existing and new CICS 3270 applications through standard WWW browsers. The benefits are threefold:

- Liberates applications

  A customer′s existing transaction processing applications, whether on the mainframe or distributed, can now be accessed from any WWW browser with no change.
- Extends the enterprise network

  In-house applications can be made available through the Internet to third parties or staff not on the enterprise network.
- Secure and reliable end-to-end communications over the Internet

  Traffic over the Internet is secured through encryption, and access to the enterprise network is controlled with a firewall. The end user can log on to CICS in the usual way, with no new procedures and no loss of security.

2.2.5 CICS for AIX V2.1.1

IBM CICS for AIX is a member of the CICS Family of Products. IBM CICS for AIX offers a highly functional, scalable platform suitable for customers worldwide to implement in production environments for their business and commercial needs.

CICS for AIX enables you to deploy mission-critical client/server applications. Thus applications based on CICS for AIX can combine the advantages of graphical user interface (GUI)-based client interfaces, such as rapid development and ease of use, with the management, data integrity, and security of an application server.

Chapter 3. CICS for AIX

OLTP systems are complex, and there are many different issues to consider if the systems are to be available when required and the data within is to be accurate and consistent. For many enterprises these issues are essential:

- Recovery
- Security
- Data integrity
- Reliability
- Availability
- Flexibility
- Operability

When a system goes down, it is essential to know that any work in progress at the time of failure can be resolved and consistency restored. Therefore each of the issues listed above must be addressed in some considerable level of detail.

For many enterprises, however, delivering the business logic only is quite sufficient. Enterprises want software that will free them to develop the logic and code for their business systems, software that will sit between the operating system and the business logic and provide a logical interface to terminals or data. A business application should not have to know the underlying file or database manager when accessing data, for example.

CICS for AIX, as an application server, provides such an application-enabling environment in an open systems environment. It presents application developers with a collection of services, a framework in which to develop business systems and yet still be able to have key features such as recovery and data integrity.

CICS for AIX is a version of IBM′s general-purpose OLTP software, CICS. It runs in the AIX operating system on IBM RISC System/6000 hardware.

Applications within CICS for AIX have access to any data that other members of the CICS family can access. Applications can be ported from other CICS family members to CICS for AIX. By implementing CICS for AIX you have access to a large pool of skilled people fluent in the use of the CICS, a key consideration for any application development project. With CICS for AIX you can build distributed applications using client/server technology, and with CICS SM you can manage those systems, building wide-ranging, complete business solutions. With CICS SM you have a tool that specifically caters for the systems and workload management of your CICS for AIX systems.

CICS for AIX provides a set of services that facilitate the writing of effective and portable application programs. The services free an application program from needing to have a detailed knowledge of the entities that it seeks to operate and manipulate. For example, the application program does not have to know the exact characteristics of a particular terminal or the precise location of a particular piece of data. CICS for AIX runs as an application server within the AIX operating system. It handles transaction processing work on behalf of the application, saving the application itself from being concerned with specialized tasks such as task scheduling and control, memory utilization, data routing, locking, and recovery.

CICS for AIX is a very flexible product that allows many possible configurations to be implemented. Thus a high degree of tailoring is available to suit individual customers and the different services of a given customer. It would be a mammoth task to list all possible combinations that CICS for AIX will support. We do not attempt to take on that exercise. Instead we show the comprehensive and flexible nature of the product.

3.1 Services

Although we explain the native CICS services in this chapter, we assume that the reader has some degree of familiarity with CICS concepts. For those readers who are unfamiliar with the CICS environment, see the *CICS Family General Information Manual*, GC33-0155-05.

CICS for AIX provides a number of services that enable an application to run in a transaction processing environment as well as have access to data stored in a number of different formats. These services are:

- Transactional services
- CICS API
- Client/server support
- Access to data

We now examine these services in more detail.

3.1.1 Transactional Services

An application in a CICS environment has available to it the key services that would be expected of any industrial-strength OLTP server, namely:

- Recovery
- Security
- Data integrity
- Reliability
- Availability

3.1.1.1 Recovery

Through services supplied by the Encina Base it is possible to initiate, participate in, and commit or back out distributed transactions. The application does not have to communicate with the Encina Base. All of the application′s interactions are with CICS. Encina Base manages the two-phase commit process between a transaction manager (CICS) and a resource manager, such as a relational database or Encina SFS. The Encina Server Core is built on the Encina Base to provide facilities for managing recoverable data, such as data that is accessed and updated by transactions. The Encina Server Core facilities includes an X/Open XA interface to permit the use of XA-compliant managers.

3.1.1.2 Security

CICS provides a number of different types of security:

- User authentication using either DCE or a CICS userid and password
- Transaction security level checking to restrict transaction start requests
- Resource security level checking to restrict access to resources such as files

- Bind-time, link, and user security for intercommunications. Bind-time is an SNA implementation. Link and user security expand the use of transaction security and resource security to the connection.

Note: When accessing a relational database through CICS using the XA interface, there are important security considerations. The granularity of database access from CICS is only at the region level. Thus all transactions that run against an XA-enabled database access that database with the userid of the CICS region. A security policy in this environment should therefore be based on CICS security facilities.

3.1.1.3 Data Integrity

Locking of data resources helps to ensure that two transactions do not attempt to update the same piece of data. Thus updates cannot be lost. The process of locking data also serves to protect data integrity in the event of a failure. Locking can have an adverse effect on performance, but the effect can be minimized by having short logical units of work (LUWs); that is, short-running transactions. For longer-running transactions, the adverse effect of locking can be minimized by taking a checkpoint frequently and thereby committing the updates performed so far. CICS provides both implicit and explicit locking facilities.

Implicit locking: The nature of implicit locking varies according to the file system used; that is, whether the files are held in an Encina SFS or in DB2 for AIX.

For a nonrecoverable Encina SFS file, Encina SFS locks the record during an update. The lock is held only until the record rewrite is performed. Because a database manager does not cope well with the concept of nonrecoverable files, all files are seen as recoverable. For recoverable files held in both SFS and DB2 for AIX, the lock is held until the end of the LUW, so that an update committed by one transaction cannot be backed out by another.

Explicit locking: CICS provides two explicit locking commands, one to enqueue and one to dequeue a resource name. The resource name is arbitrary and is chosen to be meaningful to the application. The first transaction to enqueue on a given resource name is allowed to proceed. Any subsequent enqueue requests issued before the first transaction has finished with the resource (dequeued or LUW completes) causes the requesting transactions to be suspended. For these commands to be effective, all transactions must adhere to the same rules. For example, consider the case where access to a portion of the common work area (CWA) in CICS is controlled by the use of enqueue/dequeue CICS API commands. A transaction that accesses the CWA without first enqueuing will be allowed to do so without being suspended, and the protection will be violated.

3.1.1.4 Reliability

CICS for AIX is designed for the commercial marketplace where it is recognized that reliability is a key requirement for any piece of OLTP software. Protection of CICS for AIX code is provided by allowing an installation to specify whether CICS resources are accessible or not when application code runs.

3.1.1.5 Availability

CICS for AIX is designed to provide high availability. Through the supporting commands of the product, it is possible to dynamically add resources to a running CICS for AIX region or change the state of the resources. All resources except gateway servers, SFS schemas, SFS servers, database connections, or the CICS for AIX regions environment file can be processed in this way. This processing minimizes the amount of downtime that a region experiences to make changes to existing applications or add new ones. The use of CICS SM in a suitably configured environment can mean that the end user is completely unaware of the outage of a particular region. See Chapter 8, "Using CICS System Manager for AIX" on page 111 for more details on the use of CICS SM.

3.1.2 CICS API

CICS offers a common set of programming commands that an application program can use to request CICS services. The set of commands is known as the API. As the API is common to all CICS family members, CICS applications can be moved from one platform to another. It is important to note that some commands are not available with all products. See *CICS Family: API Structure*, SC33-1007-01 for more details.

The commands are statements within the application program placed at appropriate points. CICS for AIX supports only those programs that use the command-level version of the API. The commands that are provided through the API can be categorized into four areas as described below.

Presentation services: Presentation services are used for communication between the end user and the transaction processing system. Presentation services interface with the presentation management facilities of the system, which may or may not be part of CICS.

Data services: Data services are used to retrieve and update data. Data services provide access to files, temporary storage or transient data queues, and CICS journals.

Business logic: Business logic refers to the manipulation of data from the time that it is retrieved from storage to the time it is either presented to the user or updated. This group of commands covers a range of facilities:

- Program execution services—These services enable one program to call another or another program to be run when the current program completes.
- Timer services—These services allow program execution to be started or delayed for a period of time.
- Synchronization services—These services enable serialization of access to resources.
- Storage services—These are used to manage storage use. They enable a program to reference areas of storage and obtain and free storage.
- LUW services—These services are used to delimit LUWs in a transaction.
- Configuration services— These services enable an application program to inquire about and dynamically configure CICS run-time resource definitions, such as files or transactions.
- Intersystem communication services—These services enable distributed transaction processing between a CICS region and any system supporting APPC. The services cover starting an application on a remote system,

sending data or receiving data between CICS regions, or issuing state data such as the need to abend.

- Problem determination services—These services allow a program to handle conditions as they arise or write information such as trace data or dump that will assist in debugging the problem. There are also a number of commands to assist in analyzing the performance of both the system and individual transactions.

For a more detailed description of the API commands that are available, see the *CICS on Open Systems V2.1.1 Application Programming Reference*, SC33-1569-01.

Development and debugging facilities: Application development can be a difficult and time-consuming process. CICS for AIX assists with the process by providing:

- API commands that are coded as a series of verbs, making the calls concise yet clearly understandable
- Extensive documentation that provides guidance on how to develop applications as well as an API reference
- Commands to compile programs
- Multiple language support:
  - C
  - IBM COBOL
  - Micro Focus COBOL
  - IBM PL/1
- Debugging tools:
  - A CICS transaction to turn the IBM Application Debugging facility on or off
  - A CICS transaction to enable you to interactively issue CICS API commands or obtain the syntax of those commands
  - A CICS transaction to browse the contents of as well as manipulate temporary storage queues. This same transaction can be used to access transient data queues.
  - Comprehensive CICS trace and dump facilities

3.1.3 Client/Server Support

In client/server processing a client (the requester) initiates a request to a server (the fulfiller of the request). The server may be on the same system as the client or it may be remote, that is, distributed. The roles of client and server may be played at different points in time by the same piece of software. A CICS for AIX region is capable of being a client to another CICS for AIX region and a server to a CICS on Open Systems client. A CICS on Open Systems client, however, is capable of being a client only to a CICS for AIX region because it does not have the processing capability to service a client′s request for data or an application.

A number of client/server relationships exist within CICS:

- Terminal users and CICS regions
- Encina servers and CICS regions
- DCE servers and CICS regions
- CICS regions and CICS regions

The key relationships from the point of view of implementing client/server-based, distributed applications are the terminal users and CICS regions and the CICS regions and CICS regions relationships.

3.1.3.1 Terminal Users and CICS Regions

Terminal users request services from CICS regions (servers) as clients. A CICS for AIX region can be accessed by the following CICS clients:

- CICS Client for AIX
- CICS Client for Sun Systems
- CICS Client for Windows
- CICS Client for DOS
- CICS Client for Macintosh

Therefore even in an environment where users have mixed hardware with a variety of operating systems, CICS for AIX can act as an application server. This protects the investment in existing hardware while still allowing investment in new technology. The CICS clients are discussed more fully in Chapter 4, "Using CICS Clients" on page 49.

3.1.3.2 Encina Servers and CICS Regions

CICS can use the Encina PPC Gateway for intercommunication across an SNA network. See 3.3.3, "SNA" on page 34 for more details. The Encina SFS can be used for CICS queue and file management. CICS for AIX V2.1 introduces the option of holding CICS queues and files within DB2. The Encina components are covered in more detail in Chapter 6, "Interactions with Encina for AIX" on page 77.

3.1.3.3 DCE Services

CICS can optionally be configured to use DCE security services and Cell Directory Services (CDS). The DCE security service is used to authenticate users. CDS are used to help clients locate servers. The use of DCE by CICS is covered more fully in Chapter 5, "Interactions with DCE" on page 69.

Note: Alternatively an installation may wish to use encrypted passwords for CICS userids and manually set up end-point mapping for the regions, the SFS, and any PPC gateways instead of using DCE security services and CDS.

3.1.3.4 CICS Regions and CICS Regions

CICS for AIX regions can communicate with other systems. Thus users of a local region can access services that are available on a remote system. In this case the local region is the client, and the remote region, the server. Alternatively users of a remote system can access services available on the local region. In this case the remote region is the client, and the local region, the server. This CICS-to-CICS intercommunication provides an effective and flexible base for the implementation of distributed client/server processing.

CICS-to-CICS intercommunication can take several forms. In general it is implemented such that applications or users are unaware that the resources they are using are located on a remote system. The intercommunication facilities are:

- Distributed program link (DPL)

  DPL enables a program to link to another program that resides on a different system. This can be useful where there is a large amount of data to be manipulated. The manipulation can be performed close to the data location, and the result returned to the local system. This saves having to transport all of the data to the local system.
- Function shipping

Function shipping enables an application to access certain data such as data held in files, temporary storage, or transient data queues belonging to another system.

- Transaction routing

  Transaction routing enables a transaction on a remote system to be run as though it were on the local system.

- Asynchronous processing

  Asynchronous processing enables an application program to start a program on another CICS system. The started program runs independently of the program that issued the start request.

- Distributed transaction processing (DTP)

  Distributed transaction processing is a means whereby two applications running on different systems can pass information to one another. DTP is the only CICS intercommunication facility that enables communication between CICS and non-CICS applications. The communication protocol used is APPC.

Given that it is now possible to perform updates in more than one system from the invocation of a single transaction, some form of data integrity is required to ensure that all updates to recoverable data either completed or backed out. In the interproduct communication environment that CICS allows us to have, a coordinated commitment control process is required. At logical points of consistency called *synchronization points* (*syncpoints*), transactions can exchange synchronization requests.

The APPC architecture that all CICS products can use defines three levels of synchronization. The synchronization level is established when the two communicating systems create a session. In CICS these three levels are known as levels 0, 1, and 2:

- Level 0 provides no synchronization support.
- Level 1 allows conversing transactions to exchange private synchronization requests but does not provide system-level syncpoint exchanges.
- Level 2 provides system-level syncpoint exchanges.

3.1.4 Access to Data

CICS provides access to the following data storage facilities:

- File services

  File services are used to read, update, add, delete, and browse data in local and remote files. When used locally, these commands provide access to files managed in an Encina SFS or a DB2 for AIX system. When a remote file is read, the file manager is the underlying method of the remote system. The application does not have to be aware of the file manager implementation.

- Queue services

  Queue services provide a facility to pass temporary data between tasks or programs. The data stored in this facility can persist over multiple executions of a CICS system and can represent permanent data. Two queues are available: transient data and temporary storage. Transient data queue names must be previously defined before an application can use them. Transient data queues must be read sequentially, and each item can be read only once. Transient data is used for queued data such as an audit trail and output for a printer.

Temporary storage queue names do not have to be previously defined, and they can be updated in place. Temporary storage queues can be written to a file or to memory.

For a more detailed description and comparison of these queue services, see *CICS on Open Systems Application Programming Guide*, SC33-1568-01.

- Relational database services

  CICS provides access to relational databases that provide a programmable interface through SQL commands in either COBOL, C, or PL/1. Provided that the relational database management system (RDBMS) is compliant with the X/Open XA interface, it is possible to use two-phase commit processing between the resources coordinated by CICS. See section Chapter 7, “Interactions with the DB2 Family of Products” on page 85 for more details about processing with DB2.

- Journal services

  CICS provides facilities for creating and maintaining journals during CICS processing. A journal is a set of special-purpose sequential files. An application can use a journal as an audit trail to record which users submitted which transactions or which users accessed which records in a file, for example.

3.2 Implementation

In this section we provide an introduction to the structure of a CICS for AIX region. We do not provide a detailed explanation of the internals of CICS for AIX.

A CICS for AIX region consists of:

- A set of program executables, that is, the system code required for the product, including the necessary CICS, Encina, and DCE code
- A file structure that contains definitions for the region, servers, clients, and a number of run-time files. The definitions are used to configure the system. The files holding each set of definitions are referred to as *stanzas*. The run-time files contain such items as the message log, warm-start definitions, and recovery log.
- A set of application programs provided by the enterprise that will perform the required business processing
- User data, which may be held in an Encina SFS, in a database, or on a remote system

See the *CICS for AIX Planning and Installation Guide*, GC33-1773-00, for more information about the directory structure of the regions, servers, and clients.

3.2.1 Components

The major components of CICS for AIX are:

- The CICS client processes that you use to attach to a region and through which you run transactions
- The transaction scheduler, a process that receives requests to run transactions, prioritizes and schedules them, and then dispatches them to an application server for processing

- A pool of application servers that execute the transactions and interact with the CICS client processes to send and receive terminal input and output
- The CICS queue and file manager, which is responsible for managing the CICS queues and application files
- A PPC Gateway server used when required for communication across SNA networks

3.2.1.1 Client Processes

A CICS user interacts with a CICS client. The client communicates with the CICS region processes. The client process does not have to be located in the same physical machine as the CICS region. The clients available and the protocols used are covered in Chapter 4, “Using CICS Clients” on page 49.

3.2.1.2 Transaction Scheduler

The transaction scheduler (also known as the remote procedure call (RPC) listener) is a multithreaded process responsible for scheduling and dispatching transactions to be run. It is configured with the details of all users, devices, and transactions in the region. As these change, the scheduler is kept up to date. Each transaction has an associated priority. When a transaction is submitted, the transaction scheduler computes an overall priority which it uses to prioritize requests in times of heavy demand. An incoming RPC from a client is processed on one of the threads of the transaction scheduler. The default number of threads is ten times the maximum number of application servers that is specified in the Region Definition stanza, subject to the limit of the number of threads allowed per process by the operating system. The limit on the number of listener threads is approximately 500. The exact number varies according to the number of other threads that exist in the transaction scheduler.

In most cases a transaction can be scheduled synchronously when fewer than 80% of the available concurrent RPC server threads have work to do. When the 80% threshold is exceeded, an asynchronous mode of scheduling is adopted.

The processing that takes place in synchronous mode is as follows: If there is a free application server, the transaction scheduler signals the application server that there is work to be processed, and the transaction will run immediately. If there is no free application server or the transaction cannot be processed for some reason, such as a user-imposed limit on the number of transactions that may run concurrently, the transaction is placed on a work queue to wait for a free application server.

In the asynchronous mode of operation, the RPC from the CICS client is returned with a notification that the transaction will be processed later. When a CICS application server becomes free, it processes the transaction and sends the response to the CICS client. The work queue is held within and managed by CICS. The number of *keep alive* RPCs between the transaction scheduler and the CICS client is reduced. A *keep alive* RPC occurs at 1-sec intervals and is required when there is an outstanding RPC, which there would be in this case between the client and the CICS region.

3.2.1.3 Application Servers

An application server is a multithreaded CICS process that provides a complete environment for running a CICS transaction. An application server obtains work in one of two ways: (1) the transaction scheduler signals that there are transactions to be processed, or (2) at the end of processing its current transaction it looks to see whether there is any outstanding work on the work queue.

If a transaction is conversational, it is processed by one application server, which is used by the transaction for the life of the transaction. Thus, if a transaction were waiting for terminal input and the user went for coffee, it would hold the application server, then the application server, making it unavailable to process other work. Conversational programming is not a good programming model to use. See 3.4.5, "Application Design" on page 41.

If the transaction is pseudo conversational, each transaction in turn can be processed on a different application server. When you configure CICS, you specify the minimum and maximum number of application servers for the region. The application manager always ensures that the minimum number is present. Application servers can be created and destroyed up to the maximum number defined, depending on the workload of the CICS system. All application servers have to be allocated on the same machine. The machine does not have to be the same machine as the client process, however. Application servers communicate with each other through signals and shared memory and use RPCs to communicate with other components of the system.

3.2.1.4 CICS Queue and File Management

CICS queue and file management is the function within CICS that manages the physical files, auxiliary temporary storage queues, intrapartition transient data queues, and locally managed, queued, automatic initiation requests. With the introduction of CICS for AIX V2.1, it is now possible to use a DB2 Version 2 database instead of an Encina SFS for queue and file management. The same API is used for both the Encina SFS and DB2 cases.

3.2.1.5 PPC Gateway Server

A CICS for AIX region can use one or more PPC Gateway servers to access SNA hosts. As with Encina SFS, an application server becomes a client of the PPC Gateway.

3.2.2 Processes

In this section we take a look at the actual CICS processes under AIX. Table 2 lists the CICS processes and briefly describes their functions.

Table 2 (Page 1 of 2). CICS for AIX Process Names and Functions

	Component Name	Function
cics	Main	The first process invoked
cicsas	Application server	The process on which an application runs. Typically there are a number of these processes.

Table 2 (Page 2 of 2). CICS for AIX Process Names and Functions		
	Component Name	**Function**
cicsam	Application manager	Creates the application servers and tidies up resources if an application server ends unexpectedly
cicsic	Interval control	Handles requests to run asynchronous transactions as required
cicsip	Client′s listener	Accepts CICS family TCP/IP connection requests from CICS on Open Systems clients and CICS regions
cicsld	Log demon server	Writes to the CICS log when signaled by an application server
cicsrl	RPC listener	Picks up incoming RPCs that request a new transaction to start processing. The request is placed on a work queue from which the application servers read.
cicsrm	Recovery manager	Initializes recovery as required. Creates recovery processes to recover transactions
cicsrs	Recovery server	Recovers an interrupted transaction or a crashed application server
cicssl	Local SNA listener	Accepts intersystem communication requests from local SNA services

The use of multithreading enables multiple concurrent units of work to be active within a single operating system process, providing concurrence within applications. The multiple processes and multithreading of the CICS for AIX architecture lend themselves to the exploitation of RISC System/6000 SMP hardware.

3.3 Intercommunication

In this section we examine the protocols on which the CICS intercommunication facilities are implemented. Knowing which protocols are available will be helpful in planning for a new service or establishing new communication paths between existing services.

A CICS for AIX region may well want to communicate with any or all of the following:

- Other CICS on Open Systems systems

- CICS on Open Systems clients
- IBM CICS Clients
- IBM mainframe-based CICS systems
- Remote LU 6.2 applications

To facilitate such communication, CICS for AIX supports several different communication protocols: DCE RPC, TCP/IP, and SNA. Each protocol has different characteristics and properties that make them more suitable for some purposes than for others. Below we briefly describe each protocol and its use by CICS for AIX.

3.3.1 DCE RPC

DCE RPC enables an application to make an RPC without regard to its location. A DCE RPC can be implemented over TCP/IP, a connection-oriented protocol, or over UDP/IP, a connectionless protocol.

DCE RPC over UDP/IP is used in CICS on Open Systems client to CICS on Open Systems region communication. Communication between a CICS on Open Systems application server and an Encina SFS can use DCE RPC over UDP/IP or TCP/IP. DCE RPC does not support synchronization levels for application update coordination. It can be considered more of a function call than a protocol in the way in which SNA is. The use of DCE RPC takes advantage of the flexibility that DCE CDS brings to CICS on Open Systems transaction processing.

3.3.2 TCP/IP

TCP/IP is a connection-oriented protocol. It includes conventions that allow communication between networks based on different hardware technologies. TCP/IP is widely available on UNIX-based systems as well some other systems, such as OS/2 and Windows NT. It is a simple protocol. For CICS to use TCP/IP for intercommunications, it has to provide much of the support for the data formats used to send intersystem requests as well as the security needed to protect system resources.

CICS for AIX offers a choice of two methods for using TCP/IP: CICS family TCP/IP or Encina PPC TCP/IP. One method or the other may be more appropriate for your installation.

3.3.2.1 CICS Family TCP/IP

CICS family TCP/IP enables connectivity for a CICS for AIX region to:

- CICS on Open Systems systems
- CICS on OS/2 regions
- CICS on Windows NT regions
- IBM CICS Clients

Figure 8 on page 33 shows a CICS for AIX region using CICS family TCP/IP to communicate with a CICS for OS/2 and an IBM CICS Client.

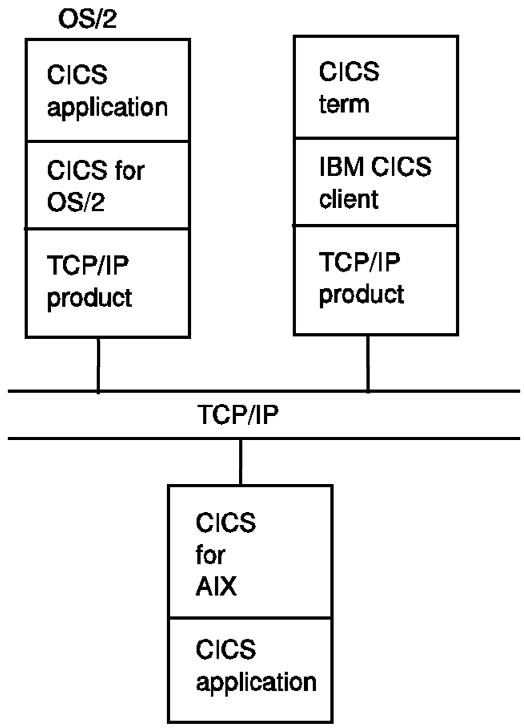

Figure 8. Interconnection Using CICS Family TCP/IP Support

When a request is first made to communicate between two systems with CICS family TCP/IP, a connection is acquired between the two systems. The connection remains for as long as the two systems are active. The connection can carry many concurrent intersystem requests flowing in either direction.

The two communicating systems do not have to be in the same DCE cell.

A CICS for AIX region can be configured to accept TCP/IP connections on one or more TCP/IP ports in the local machine. It can also receive requests on one or more token-ring adapters. This is important as the maximum number of concurrent connections through a single port is around 450. The exact number depends on the number of open file descriptors in CICS.

CICS family TCP/IP supports function shipping, transaction routing, DPL and asynchronous processing. It does not support distributed transaction processing. CICS family TCP/IP supports synchronization level 0 and level 1 for all flows.

3.3.2.2 Encina PPC TCP/IP

Encina PPC TCP/IP supports function shipping, transaction routing, DPL, asynchronous processing, and distributed transaction processing between CICS on Open Systems systems and enables distributed transaction processing with Encina applications. If DCE CDS is in use, both systems must be in the same DCE cell. If the two communicating systems are not in the same DCE cell, consider using CICS family TCP/IP.

Figure 9 on page 34 shows the Encina PPC Executive being used to interconnect CICS for AIX regions and an application that is using the Encina PPC Executive.

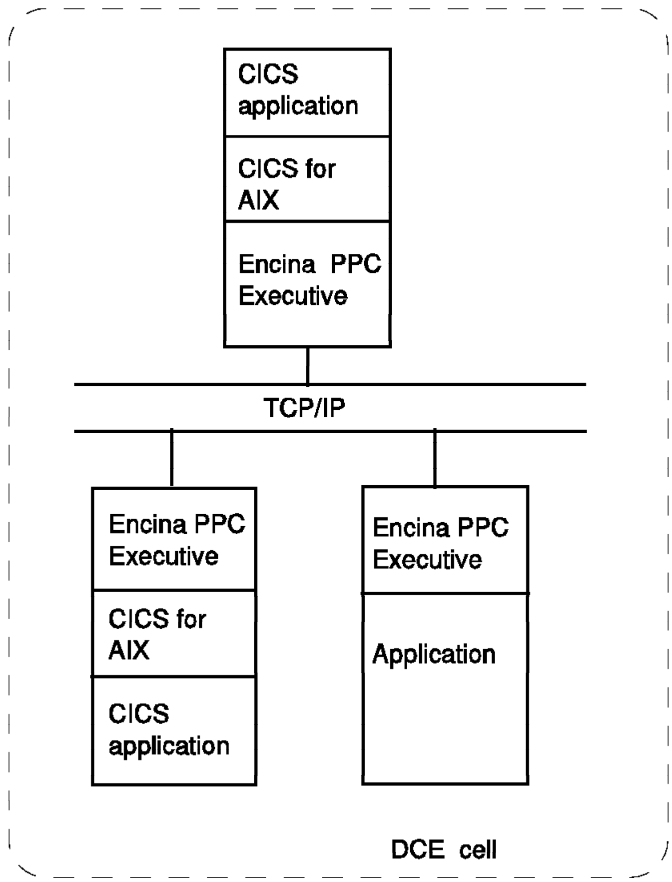

Figure 9. Interconnection Using Encina PPC TCP/IP Support

When an intersystem request is made, the name service is used to locate the required region. A connection is then set up between the two systems and used exclusively by the intersystem communication requests. The connection is closed down when the request has completed. Depending on the frequency of the requests, a lot of session startup and shutdown processing could be involved. If this is the case, consider using CICS family TCP/IP where the connection remains for as long as the two systems are active. Synchronization levels 0, 1, and 2 are supported with Encina PPC TCP/IP.

3.3.3 SNA

With SNA, proprietary network architecture from IBM, the end points of conversations operate as logical entities, or a logical type of device. Thus, a piece of code can communicate in the same manner as an actual device and so appear to its partner in the conversation to be a device. SNA is widely used in IBM mainframe-based systems. It is also now available on a number of UNIX-based systems such as AIX. It is implemented on these systems through products such as SNA Server for AIX. The availability of SNA on these UNIX-based systems facilitates communication between the UNIX-based systems and applications that run on mainframes.

CICS for AIX regions can communicate across SNA with any APPC-capable system, including:

- Other CICS on Open Systems regions
- IBM CICS clients
- CICS for OS/2 regions

- CICS for Windows NT regions
- CICS/400 regions
- IBM mainframe-based CICS regions

CICS for AIX offers a choice of two methods for using SNA: local SNA and the Encina PPC Gateway. Both methods support all of the CICS intercommunication facilities.

3.3.3.1 Local SNA

This method provides the fastest connectivity. A CICS for AIX region can communicate with every other member of the CICS family through this method. Use of local SNA requires that an appropriate SNA product, such as AIX SNA Server for AIX, be installed on the same machine as the CICS for AIX region. Synchronization levels 0 and 1 are supported.

Figure 10 shows CICS for AIX using local SNA support to communicate with other CICS systems and other APPC applications.

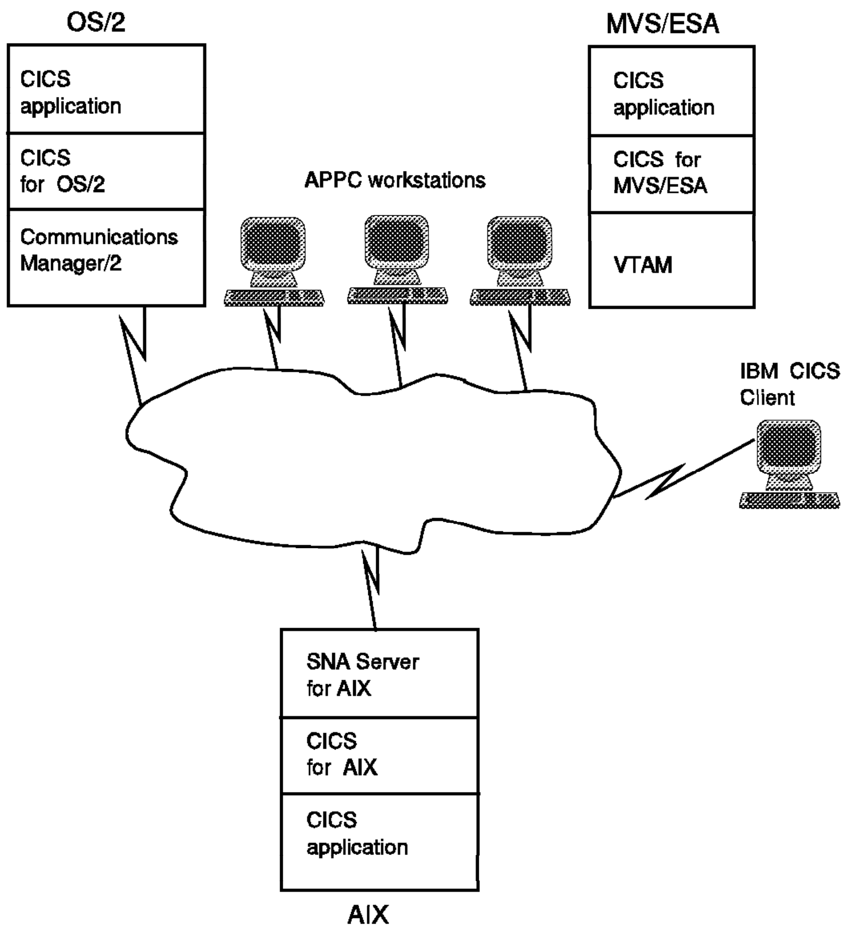

Figure 10. Interconnection Using Local SNA Support

Encina PPC Gateway: With this method CICS communicates with the Encina PPC Gateway through TCP/IP, and the Encina PPC Gateway provides a link to the SNA network. The Encina PPC Gateway can be in the same machine as the CICS for AIX region or in a different machine. If DCE CDS is in use, the Encina PPC Gateway and the CICS for AIX should be in the same DCE cell. Figure 11 on page 36 shows a CICS for AIX region using the Encina PPC Executive to connect to a PPC Gateway server machine, which in turn connects to an SNA network.

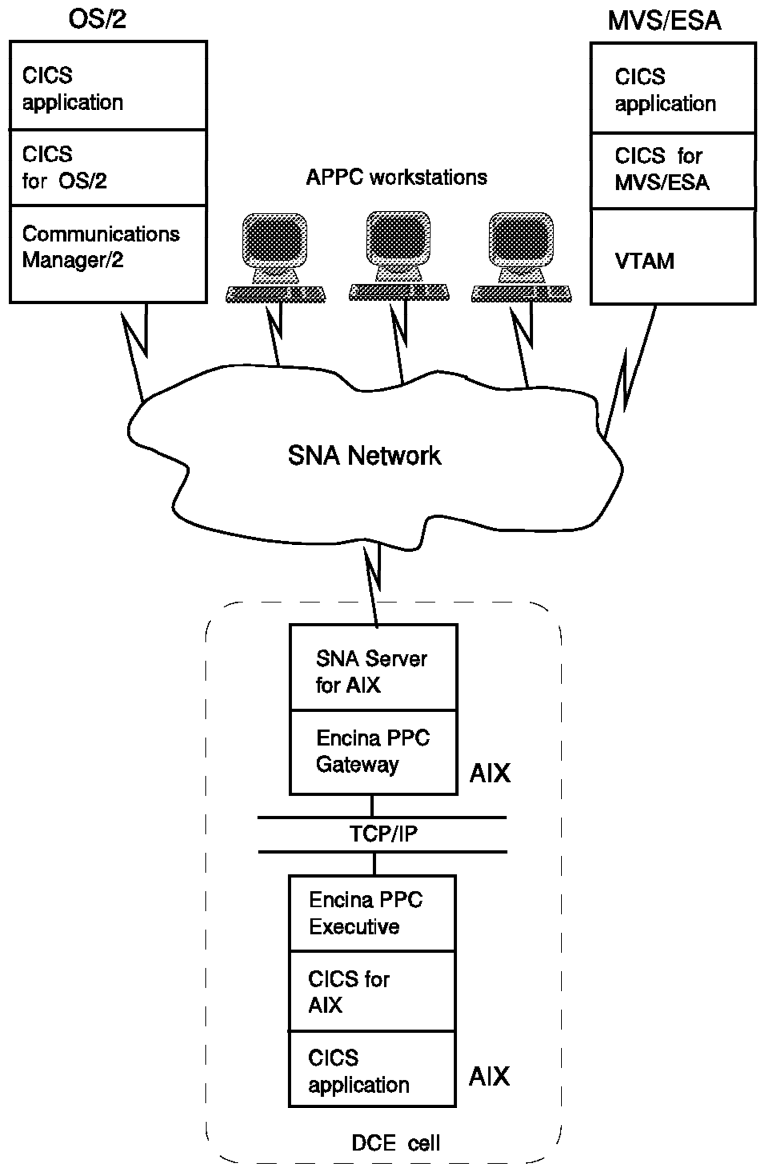

Figure 11. Interconnection Using Encina PPC Gateway Support

The Encina PPC Gateway uses a product such as SNA Server for AIX to connect to the remote SNA system. SNA Server for AIX has to be installed on the same machine as the Encina PPC Gateway.

A CICS for AIX region can use more than one Encina PPC Gateway. The Encina PPC Gateways must be on the same system. Reasons for using more than one Encina PPC Gateway might be:

- To allow the network links from a large number of remote systems to be spread across more than one gateway machine and hence across multiple SNA products
- To introduce some level of redundancy so that a backup is available if one Encina PPC Gateway fails
- To spread the processing load over more than one Encina PPC Gateway

An Encina PPC Gateway can be used by more than one CICS for AIX region if the CICS for AIX regions do not make too many SNA intercommunication requests. Be careful, however; the Encina PPC Gateway is implemented on a single process within the operating system and could become swamped with work. Problem determination is more difficult if more than one CICS for AIX region uses an Encina PPC Gateway.

3.3.3.2 Mixing TCP/IP and SNA

Using the products available, you can mix the methods used to connect your CICS systems to meet complex interconnection requirements. For example, a CICS for AIX region could use the Encina PPC Executive to communicate with another CICS for AIX region, while also communicating over SNA using both local SNA support and an Encina PPC Gateway server.

Figure 12 on page 38 shows a CICS for AIX region communicating with another CICS for AIX region using TCP/IP. Encina TCP/IP is used because both regions are in the same DCE cell. The CICS for AIX region is also communicating across an SNA network to a CICS for OS/2 system, some APPC workstations, and a CICS for MVS/ESA system. The communications between the CICS for OS/2 system and the workstations use local SNA support. Local SNA was chosen because CICS for OS/2 and the APPC workstations do not support synchronization level 2. The communications with CICS for MVS/ESA use a PPC Gateway because synchronization level 2 is required. The PPC Gateway server is running on the same machine as the CICS region and is using the same SNA product.

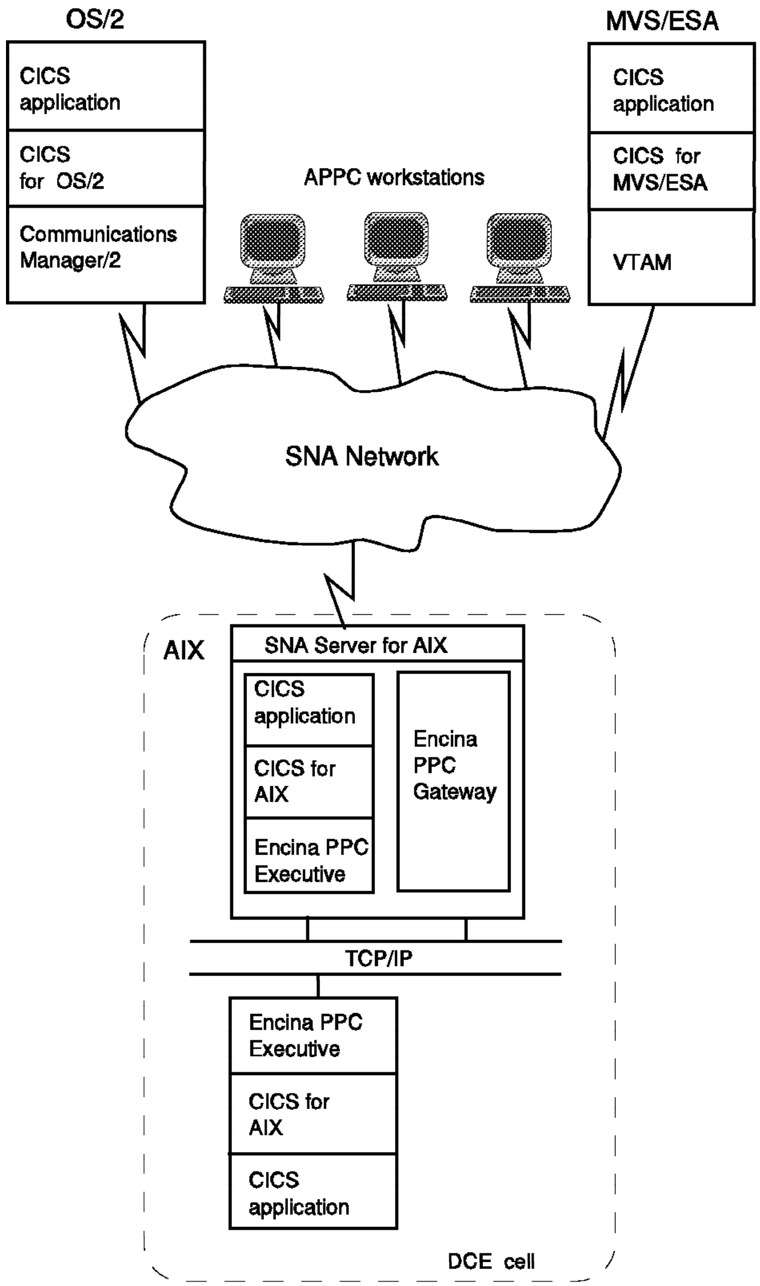

Figure 12. Interconnection Using TCP/IP, a PPC Gateway Server, and SNA

3.4 Planning Considerations for CICS for AIX

In this section we examine key configuration issues when implementing an OLTP system using CICS for AIX, with a focus on obtaining optimal performance. It is important that you understand that system configuration is the first part of system tuning. The way in which CICS for AIX is configured can have a fundamental effect on the resulting performance that can be obtained from an implementation. For example, consider two implementations. The first system consists of a CICS for AIX region that has all data available locally; a subset has been copied from the mainframe system. The other system consists of one CICS for AIX region that function ships requests to a mainframe CICS system. There are no constraints on either system. The first system inherently provides much better performance than the second because the latter has to remotely access the data. There may be perfectly valid reasons for setting up a system in the manner of the second. However, it cannot perform identically to the first system because of the additional processing and network delays. This example demonstrates that no amount of system tuning can overcome a poor initial design for either the CICS for AIX configuration or the application.

Although the focus in this section is on CICS, you should remember that there are other components in a system, such as an RDBMS or an interconnected CICS system. Adding a new application not only affects the CICS system in which it runs but also these related components.

You must put a lot of thought and effort into the early planning stages, and consult widely, talking to people from operations, CICS systems programming, database administration, application design, capacity planning, and user support groups. Doing this will greatly improve the likelihood that the configuration will work as intended, with the expected results. It is essential to think through the proposed configuration from the user interface to how operations will run and provide problem determination for the system. If the proposed region is to provide a service for multiple applications, you must walk through the configuration for each application to ensure that all of its needs are met.

Sometimes in implementing an aspect of a CICS system or an application, options are not available. You are forced to use a certain type of CICS client because the users already have a PC installed running OS/2 rather than an X-station, for example. In such instances it becomes a case of determining whether the function is available in the product to support the need rather than trying to evaluate the merits of the alternatives.

3.4.1 Overall Objectives

Have a clear view of what is required from the new configuration, what the primary and secondary objectives are. Decide whether system throughput, user response time, or system availability is most important. Whatever your requirements, be clear about them.

3.4.2 Capacity Planning

Where new CICS regions are being introduced as part of the introduction of a new application or where a new application is being added to an existing CICS region, it is important to undertake some capacity planning. Understand the processing requirements of the new application before you discover that it is flooding your existing configuration and could affect users of other applications. This work will help you understand which machine resources, such as CPU and disk space, are required. It will also give you something to compare the actual results to later on. This is important for a completely new application for which historical information is not available.

Before beginning your capacity planning:

- Identify the different types of business transactions
- Understand the ratio of CICS transactions to business transactions
- Determine the likely and maximum transactions rates
- Understand where the application data will be held
- Develop a profile for each type of transaction:
  - Map out the CICS services that will be required
  - Identify the recoverable resources

  - Understand how application data will be accessed from within the application. This may not be determined by CICS but by an RDBMS. It is still important to understand though
  - Understand the amount of logical and physical I/O
  - Understand the level of logging involved
- Determine which other software components, such as an RDBMS or a remote system, are involved. What is the projected load of the new applications? Can they accommodate the load?

Capacity planning will enable you to see whether the expected work can be accommodated by the current processing power and I/O capacity or whether additional machines will be required. If new machines are required, you have to determine their size. There are extremes of effort in performing capacity planning. A minimal level of effort would be to look at the application design and estimate the volume of I/O and the CPU requirements. The other extreme would be to model in quite some level of detail, the clients, the CICS for AIX region, the RDBMS, and possibly the network load.

Consider the other processes that may be competing with this CICS for AIX region; for example, is a large amount of batch work processed on the same machine?

3.4.3 Configuration

To implement new CICS systems, you first have to form an idea of what the configuration is going to be, that is, you have to know how many regions on how many processors are required. The decision may or may not be straightforward. It is likely to be influenced by a number of factors:

- The likely processing power required. From your application capacity planning, you can determine whether you will have a single or multisystem implementation.
- The design requirements of the proposed systems. If high availability of the application is a key requirement, consider implementing multiple regions capable of running the application and managing the workload. See 8.5.1, “Maximizing the Benefit” on page 121 for more details of this approach.
- If the new application is incompatible with existing applications or has special operational or security requirements, you will have to implement another CICS region.
- If the aim is to exploit a parallel database, the recommended approach is to have multiple CICS systems capable of accessing the database (see 8.5.1.3, “Using an SP Processor” on page 123).

3.4.4 Managing the Work

In implementing new CICS systems, you may find that applications are split over more than one CICS system for reasons of availability or because the processing capacity required exceeds that available from a single machine. Incoming user transactions have to be distributed over the available CICS systems. The distribution of work is an awkward problem that must be resolved. You must understand how you are going to deal with it in your environment.

The incoming work must be distributed in such a manner that no one CICS system is flooded with more work than it can process. One approach might be to distribute the number of users evenly over the available systems so that groups of users access particular systems. The allocation of work is performed by controlling the number of users who access the system. Some groups of users may be more productive than others, in which case there may be more users attached to some systems than others. When the capacity of one system is reached because a department takes on more work, for example, some users will have to access a different system in order to redistribute the work. Such an approach is inflexible and poor from the point of view of users, who should be isolated from the implementation. An improvement on this approach would be to have users access a routing region. There would be only one single system to which to connect. Work could be routed evenly across the application-supporting regions by selecting a region at random. Given a good enough random selection process, the number of requests should balance out over the available systems—assuming that all transactions generate an equal volume of work, which is not necessarily true. Some systems could become flooded with work, and as a result users will experience erratic response times.

A great improvement would be to have the intelligent routing of work requests. On the basis of the existing load and response times that transactions were receiving from a CICS region, a decision could be made as to the best system to which to route a piece of work. This approach should ensure that work is evenly distributed over the available systems and that response times are more even as one system will not become flooded. In addition users would not have to be moved from one system to another to balance the work out. This is the approach that CICS SM Workload Management offers. See Chapter 8, "Using CICS System Manager for AIX" on page 111 for more details.

3.4.5 Application Design

Efficient application design is an important component of overall system performance. Unfortunately it is one aspect of system implementation that does not receive the focus that it should although it can be one of the biggest contributors to poor performance. Applications that retrieve excessive amounts of data, for example, not only perform more slowly but also place increased demand on the rest of the system, such as the data manager and the I/O subsystem, thus impacting other applications attempting to share the same resources.

Consider the programming model for the applications, that is, pseudo conversational or conversational. A pseudo conversational model is preferred because it acquires resources for shorter periods of time. It does not hold a CICS application server while awaiting a reply from a user. Look at how programs are to intercommunicate and consider which of the CICS-provided scratch pad facilities are most suitable.

Become familiar with the *CICS On Open Systems Application Programming Guide*, SC33-1568-01, particularly the chapter on designing efficient applications. Aim to create applications that avoid affinities. This approach will facilitate future expansion, should you have to add more CICS regions to support an application. *CICS/ESA Dynamic Transaction Routing in a CICSplex*, SC33-1012-00, contains useful guidelines on avoiding transaction affinities in applications.

3.4.6 Intersystem Communication

CICS provides good intercommunication facilities. However, you should consider whether CICS systems really do need to be interconnected. There may in practice be no real decision to be made because the issue is forced in some way—for example, data on a remote system has to be accessed by user transactions, and it is infeasible to replicate the data.

If intersystem communication is used, it is sensible to optimize its use from several viewpoints:

- Performance

  The fewer calls to remote systems, the less processing work to be done. Each time there is communication over a network component, there is an inherent delay no matter what the speed of the actual link.
- Complexity of the configuration

  With an increased amount of intercommunication, the overall CICS environment becomes more complex, and determining whether all of the links are working becomes less straightforward.
- Ease of problem determination

  In a highly interconnected environment, it can become difficult to trace a problem as there are many components interacting. Problems of poor performance become more difficult to track down. It becomes difficult to capture all activity related to one system when that system may be connected to three or four others. Users will report a problem, but they will have no idea where in the complex the problem actually is.

3.4.6.1 Selecting a Protocol

When selecting a protocol for the implementation of intersystem communication, ensure that it enables you to connect the required systems and provides the required level of synchronization. For example, CICS over TCP/IP is easy to set up and has advantages in that the session it creates remains established while both systems remain active. It does not provide synchronization level 2 support though. If your application actually requires synchronization level 2, you must use Encina PPC TCP/IP or Encina PPC Gateway.

In the case where a CICS for AIX region is running multiple applications, it may be necessary to have several communication protocols in use between the local region and other CICS regions. This situation is to be avoided where possible, however, to keep the configuration as simple as possible.

Table 3 summarizes the protocols supported in CICS for AIX.

Table 3 (Page 1 of 2). Communication Methods Used by CICS across TCP/IP and SNA

	Best for	**Restrictions**
CICS family TCP/IP	Communication at synchronization level 0 or 1 with CICS on Open Systems, CICS on Windows NT, CICS for OS/2, and IBM CICS Clients across TCP/IP	DTP is not supported. CICS user security must be configured with care over these connections as it is not possible to reliably authenticate (identify) the remote system.

Table 3 (Page 2 of 2). Communication Methods Used by CICS across TCP/IP and SNA		
	Best for	**Restrictions**
Encina PPC TCP/IP	Communication at synchronization levels 0, 1, or 2 with other CICS on Open Systems regions or Encina PPC applications	If your region is using DCE CDS as a name service, the remote region or application must be in the same DCE cell as the local region.
Local SNA	Fastest synchronization level 0 or 1 communication with remote LU 6.2 (APPC) systems. These connections can be used to connect to any CICS product.	A supported SNA product must be installed on the same machine as the CICS region.
Encina PPC Gateway	Synchronization levels 0, 1, and 2 communication with remote LU 6.2 (APPC) systems. These connections can be used to connect to any CICS product.	When using AIX SNA, Encina PPC Gateway must be installed on a machine along with a supported SNA product. If your region is using DCE CDS as a name service, both the PPC Gateway and your local region must be in the same DCE cell.

Understand how intersystem sessions are created. The methods vary with the protocol. In an environment where there are frequent requests between the two systems, this session setup and shutdown activity can represent an unnecessary overhead, which could be avoided through the selection of an appropriate protocol. For example, with CICS family TCP/IP, a session is created between the two systems and stays active for as long as the two systems remain active. Encina PPC TCP/IP, however, creates a session solely for the duration of the request.

3.4.7 Choosing an Intercommunication Facility

CICS provides a number of different intercommunication facilities that complement each other. One facility will usually be more suitable and efficient for a given application than another. Below we consider the attributes of each facility and describe when it may be best used.

3.4.7.1 Function Shipping

Function shipping provides an easy means of accessing remote data resources, such as files or temporary storage queues. Each request is sent separately to the remote system, however. So, depending on the amount of data to be accessed, it may be more efficient to use a DPL or a DTP request.

Consider the case where a record is updated on a remote system. There will be one round trip to the remote system to get the record, a second to rewrite, followed by a third at syncpoint time, a total of three round trips to the remote system. A DPL program could have completed the request in a single trip by being called so that it performed synchronization processing prior to its return. The DPL case does require programming effort to implement it and there are recovery considerations to take into account, such as, What if the DPL program committed its update and there is a failure on the local system? So the choices are not always clearly defined. It is a case of looking at all of the options and choosing the best one for your environment.

3.4.7.2 Transaction Routing

Transaction routing does not have several alternatives like function shipping has. So the case for using this method is more straightforward. If you need access to a remote transaction, use transaction routing. With this facility, you do not have to implement the application and data in two places or access two systems.

3.4.7.3 Asynchronous Processing

With asynchronous processing the remotely started transaction does not affect the application that started it. This facility is useful when it is necessary to start a transaction without delaying the local transaction at all.

3.4.7.4 DPL

DPL is best used where there is a large amount of data to be manipulated on a remote system. The DPL program has local access to the data, performs the necessary manipulation, and returns the result. This process can be considerably more efficient than having to function ship each item of data.

3.4.7.5 DTP

DTP is the most flexible intercommuncation facility. The programs can send and receive as much data to each other as required. It is the most complex to set up, however, as the originating program is responsible for setting up the communications with the remote system, sending and receiving the data, and closing the communications down when it is finished.

3.4.8 Data Location

As a general rule, no matter where the data is located for an application, it is wise to keep the number of recoverable resources to a minimum. This reduces the amount of logging that has to be done to protect the updates to the recoverable resources. Reducing the number of recoverable resources may also reduce the number of participants involved in any syncpoint processing, thereby improving performance.

Application data can be held in Encina SFS, in an RDBMS, in CICS queues, or in a remote system. There are significant performance considerations for each of these methods. The choice is not always a free one and may to a large extent depend on where data is currently held. If is not feasible to move data from one platform to another, the data has to be accessed remotely.

Encina SFS: Data held within an Encina SFS has often been migrated from an IBM mainframe environment where the data was held in VSAM. The Encina SFS can be located on the same system as that of the CICS for AIX region. The advantage of doing this is that the fast local transport (FLT) method can be used between the CICS application server and the Encina SFS. This method provides a performance improvement over the use of the DCE RPC.

If the Encina SFS and CICS systems are distributed, it may be possible to use two smaller machines to provide the necessary processing power. As long as the Encina SFS and the CICS region are within the same DCE cell, there is no problem in distributing the components.

Note: Distributing the components may impact the performance of accesses to CICS files or queues held within the Encina SFS.

RDBMS: A key decision when designing a CICS application that connects to an RDBMS such as DB2 for AIX is to understand the types of connection that are available and the level of recovery that each provides. In choosing the type of connection, there is a tradeoff between the level of recovery and performance. With CICS for AIX and DB2 for AIX, there are two possible ways in which to connect to the database: using the XA interface or using the XA single-phase commit optimization.

The XA interface connection provides a full two-phase commit protocol. This protocol is required where there are updates in two or more locations and the updates have to be synchronized. Remember that updates to CICS files such as temporary storage when they are held in an Encina SFS will count as one location (although it may not be immediately obvious), as will an update on a remote CICS system that was performed at synchronization level 2.

The XA single-phase commit optimization can be used where full two-phase integrity is not required and higher performance is required. There are restrictions imposed though as a result of selecting this optimization: only one DB2 database can be supported, and it is not possible to support an XA-enabled DB2 database with a database defined under the XA single-phase commit optimization.

CICS Queues: As we discuss in 3.1.4, "Access to Data" on page 27, CICS provides transient data and temporary storage as possible facilities for the storage of permanent data. These two queue are quite different. One or the other of them may be more suited to your particular requirements. However, the use of these queues to store permanent data is not particularly recommended because the data is held in a CICS internal format, with the exception of extrapartition transient data. Thus the data is difficult to access other than from within CICS. The CICS queues may be held in either an Encina SFS or a DB2 for AIX database, depending on how you configure CICS for AIX file control.

Remote System: When data is held on a remote system, look carefully at how that data is accessed. To determine which is the most efficient intersystem communications facility to use to access that data, see 3.4.7, "Choosing an Intercommunication Facility" on page 43.

3.4.9 Selecting a Client

A number of different CICS clients are available (see Chapter 4, "Using CICS Clients" on page 49). A client based on the external call interface (ECI) rather than the external presentation interface (EPI) is the preferred client type.

3.4.10 Availability

Consider the planned hours of availability for the new system and how they might change over time. True online systems, as opposed to those accumulating a transaction file to be applied during a batch update, are increasingly common. They often have the requirement for high availability. Consider whether the proposed system and application will be able to support a 24 hours a day, 7 days a week operation, how the application updates will be performed, and which events will require a recycling of the CICS for AIX region or the RDBMS. You should also look to see what support the proposed RDBMS provides for availability. Consider whether your preferred choice of RDBMS will inhibit the level of service that has to be provided. If some applications within a CICS region have different hours of availability from others, consider how you plan to cope with this situation. This is where CICS SM can play a significant role. By having multiple configurations reflecting the different applications or resources as required at different times of the day or week, changes can easily be introduced. The changes are made by working with the CICS SM GUI rather having to issue CICS commands or log on to the CICS system directly.

You want to minimize the amount of forced downtime for maintenance or housekeeping procedures. Adopting the approach we discuss in 8.5.1.1, “Combining Resources” on page 121 can help avoid such outages.

3.4.11 Operations

Consider how the operations staff will run the new systems on a day-to-day basis. The only performance recommendation here is to ensure that the staff has clear guidance and effective tools with which to manage the system because a system or application that is down has zero performance.

Again CICS SM can play a major role in improving the ease with which the CICS for AIX regions and their resources are operated. By having all necessary configurations defined, the operations staff can simply ensure that the relevant configuration is active at the required time by working with the CICS SM GUI. The single point of control (SPOC) becomes a reality and a benefit. The single system image (SSI) allows operations staff to determine the state of any CICS resource without having to access the actual machine. As well as providing a more effective means of managing the environment, CICS SM also provides a more efficient means.

3.4.12 CICS Testing and Evaluation

Testing should be a key element in the introduction of a new system or application. We recommend taking the time to develop test tools. The tools do not have to be very complex. Consider using an ECI- or EPI-based application. The use of a multithreaded client will allow more users to be simulated for a given amount of machine resource.

Testing is often not given the focus that it deserves. It is, however, the last opportunity to check the integrity and function before the actual users start using the system. Problems discovered in a production environment are far more difficult to resolve where there is greatly increased pressure for the service to made available. Volume testing will enable you to gain confidence that the system can cope with the expected levels of work. It will also enable you to stress the system to see what happens if, for example, 110% of the expected volume of work is placed in the system.

When you have the ability to drive work through the system, it is wise to introduce some way to evaluate user and system performance, such as measuring user response times, or CPU usage, to give you confidence that the system is performing as expected. User and system performance evaluation can indicate that the system is not tuned properly—perhaps the database buffers were not set correctly and twice the expected level of I/O is taking place, or perhaps a trace was left on after debugging some application problems.

3.4.13 Security

Consider the level of security that you need. CICS provides facilities for user authentication and controlled access to transactions and resources. If these facilities are sufficient, you can consider running without the use of DCE security service. However, if you think the CICS facilities provide an inadequate level of security, use the DCE security service to provide additional levels of security. See 5.1.3, "DCE Security Service" on page 70 for more details.

Review each of the components of your system. It may be appropriate to have different levels of security in different places. For example, connection attachment security for sessions between the servers in a group of CICS regions may be sufficient if they are on a secure LAN. For a remote cicsterm or region, a higher level of security may be appropriate. The level will depend on the perceived risk of electronic eavesdropping, as well as the sensitivity of the data being transmitted.

3.4.14 Ongoing Monitoring

Ongoing monitoring can assist in keeping the operations support team informed as to how well the system is performing. Up-to-date and accurate information on the state of CICS systems enables correct decisions to be made in the event of any problems—and before users are fully aware of any impact. Thus it is important to have some means of knowing on an ongoing basis the status of the CICS systems and resources. The provision of this information must be planned. It will not just happen. In a large OLTP environment, monitoring must be automated. Individuals cannot monitor at the level of detail and scope that are required.

CICS SM yet again can add significant value to the operation and management of your CICS systems and resources. Through its event notification facility, the status of key CICS systems or resources under its management can be effectively monitored. Such monitoring can provide immediate notification of the change in the state of a resource. A screen located in the operations area provides staff with the ability to diagnose problems and react rapidly.

CICS for AIX also provides monitoring facilities: statistics at the system level, and monitoring at the individual transaction level. Statistics aid the management of resources such as files or queues. They also provide a useful snapshot of peak activity so that you can see, for example, whether a storage limit is about to be reached. Statistics collected on a regular and ongoing basis can provide you with an understanding of the long-term behavior of your CICS systems. They can assist in predicting behavior and thereby avoiding unpleasant surprises: Is the volume of work steadily increasing or decreasing? Are the peaks as intense as they were? Are the peaks likely to cause a problem shortly?

The CICS monitoring facility provides a high level of detail and consequently can significantly increase the time it takes to run CICS transactions. The monitoring facility should be used with caution. Third-party offerings are available to monitor CICS for AIX systems on an ongoing basis. They may well work on a sampling basis. They will most likely have a lower overhead than using the CICS monitoring facility, although the level of information that they provide may not be as detailed.

3.5 Summary

CICS for AIX provides an excellent basis for reliable, secure, industrial strength OLTP in an open distributed systems environment. CICS for AIX has the advantage of being a member of a family. Thus it can communicate with other members of the CICS family to access data that they may hold or have access to. CICS for AIX provides support for clients on multiple operating systems; thus it can be the OLTP server where users operate from a variety of hardware and operating systems. CICS for AIX provides a wide and varied API that is available in multiple programming languages. It has access to multiple types of data. It provides multiple forms of intersystem communication. It allows connectivity to non-CICS applications if a suitable protocol is used.

The key characteristic of CICS for AIX is flexibility. CICS for AIX recognizes that the world is complex. Its aim is to simplify, so that application development can see the required transactional services as a set of services to be called. Thus application developers can concentrate on their core objective, producing business applications to best serve the enterprise.

Chapter 4. Using CICS Clients

Users require an interface to interact with a CICS for AIX region. The longer-established members of the CICS family, those running on mainframe computers, were designed with a terminal network that largely consists of 3270-type displays and printers. Tens of thousands of CICS applications have been written using the 3270 data stream for terminal input and output. The existing investment in these CICS applications must be preserved and protected. CICS for AIX and CICS Clients address this requirement and enable such CICS applications to run successfully without any changes. The CICS Clients are designed with modern requirements in mind, such as the ability to add a GUI to a CICS Client or the ability to access a CICS for AIX region from a non-CICS application.

CICS Clients support multiple functions, and multiple versions of CICS Clients are available. By *multiple functions* we mean that the CICS client produced for one operating system can provide a number of different interfaces, giving you greater flexibility in the type of user interface that you decide to implement. By *multiple versions* we mean that CICS Clients are available to run on multiple, different hardware types and operating systems. This increased choice protects the investment in the hardware and software that your users use.

The design of CICS Clients is such that the client accesses the business application that runs in the server machine. Typically the client will run on a different computer from the server so that processor usage is offloaded from the server machine.

CICS Clients enable users to access CICS for AIX systems to run their business applications. Through CICS Clients you can interact with the CICS for AIX system as a terminal (3270-emulated terminal) user. CICS Clients also provide additional functions that enable you to write your own user interface while still providing client access to a CICS for AIX region. These functions are the ECI and the EPI. The availability of the ECI and EPI greatly extends the choice when deciding on a user interface. If none of the standard CICS-provided interfaces is suitable, you can create your own and combine it with the ECI or EPI to provide the CICS client component.

4.1 Application Server Model

The design of CICS Clients is based on the application server model, a processing model used in an open distributed multivendor OLTP environment. The purpose of the model is to separate the presentation logic from the business logic to enable the client to run on a separate processor from the business application and so offload processor usage from the server machine.

Figure 13 on page 50 illustrates the components in the application server model. In this model the screen handling is run on the end-user workstation, or client computer. The business application is run in an application enabling environment on a server computer located on the LAN. The application is accessed through a thin layer of client code in the client computer.

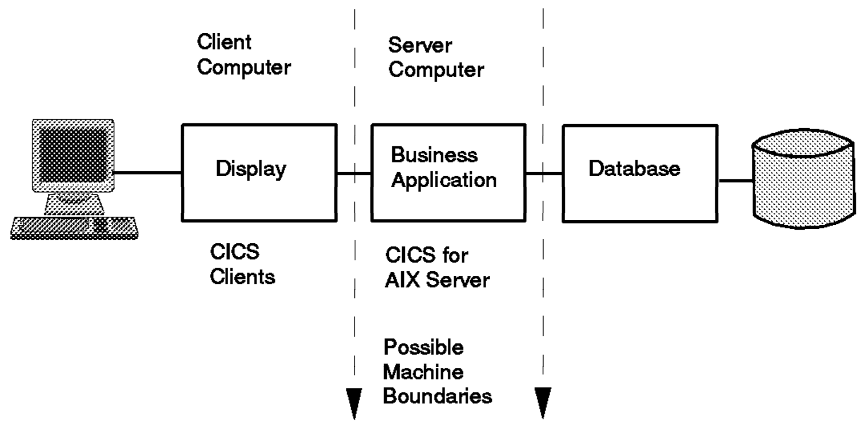

Figure 13. Application Server Model

Let us relate this model to the CICS environment. The client computer runs a CICS Client, and the business application runs in a CICS for AIX region on a server computer on the LAN and is therefore not present in the client computer. This approach ensures that the business application is in a trusted environment, well away from the client computer. Resources used by the business application are accessed through the CICS for AIX region. Thus the various application services that CICS provides are available across the entire application, encompassing all of the various resources being used. If the business application, for example, changes a database and a flat file in the same business step, CICS for AIX can ensure that the changes are kept in synchronization, and integrity is thus maintained across these different resource types.

As the load increases, the application server model enables multiple application servers to be set up on the LAN, and those servers can also provide gateways to other connected systems.

The application server model offers the maximum flexibility and the best level of ″future proofing″ while providing an environment with excellent integrity and performance characteristics.

4.2 Architecture of CICS Clients

The same architecture is used across all CICS Clients (see Figure 14 on page 51). Understanding the architecture should help you to visualize how the clients operate, so you can select the most appropriate client and design your interfaces accordingly.

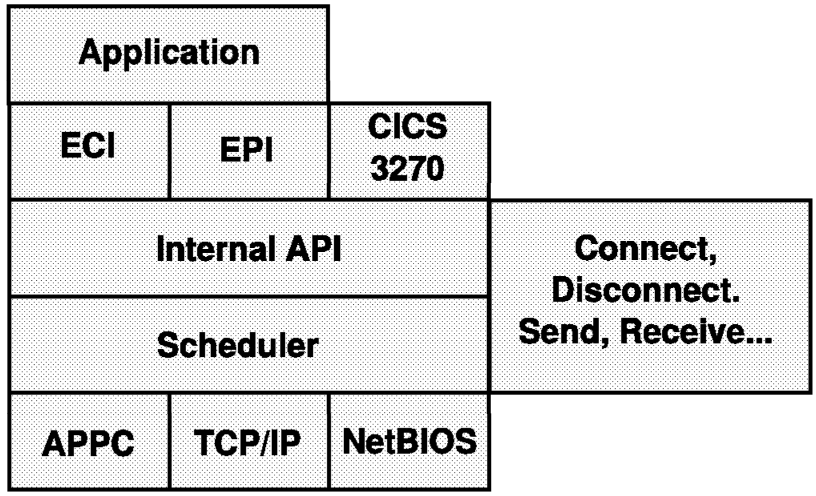

Figure 14. CICS Client Architecture

The components of the architecture are:

Application Your user application using the ECI or the EPI

CICS 3270 Support for your CICS 3270 terminal and printer emulators (see 4.2.3, "CICS 3270 Terminal Emulator" on page 53 for more information)

Internal API CICS Client internal API for controlling client functions

Scheduler Scheduling of client tasks

APPC, TCP/IP, NetBIOS Communication protocol code

Connect, Disconnect. Connection and disconnection of clients and client terminals

Send, Receive... Sending and receiving of data to and from the server

There are a number of different implementations of the architecture. Each is designed to achieve a different function. The implementations available are:

- CICS ECI
- CICS EPI
- CICS 3270 emulator
  - cicsterm
  - cicsteld

The ECI and EPI provide a programmable interface that enables CICS client processing to be accommodated in a non-CICS application program or a more tailored user interface than the CICS 3270 emulator can supply. The CICS 3270 emulator versions provide a means of displaying a 3270 data stream. Using a CICS 3270 emulator, a user can interact with CICS for AIX applications. The format of the display is restricted to 3270 display characters, however. Let us now examine each of these implementations in more detail.

4.2.1 CICS External Call Interface

With the CICS ECI, CICS applications are designed with the business logic on the CICS server and the presentation logic on the workstation-based CICS Clients. ECI enables a non-CICS client application to call a CICS application, synchronously or asynchronously, as a subroutine. The client application communicates with the CICS server program using the CICS communication area (COMMAREA). The CICS program typically populates the COMMAREA with data accessed from files or databases, and the data is then returned to the client for manipulation or display.

The ECI is a remote call from a user′s application on the client workstation to a CICS program on a server. At the CICS server, the ECI looks like a DPL from a partner CICS system.

CICS Clients, with the exception of CICS Client for DOS, can support an unlimited number of outstanding, concurrent ECI calls to a CICS server, with no restrictions on communication protocols or functions or whether the calls are to the same or a different CICS system. In the case of CICS Client for DOS, there is a limit of 16 concurrent calls.

Using ECI, any program on the workstation can access facilities on the CICS for AIX server. The ECI provides maximum flexibility for the client-server environment because of the ability to change the client without affecting the server and the server without affecting the client.

Figure 15 shows the CICS ECI model.

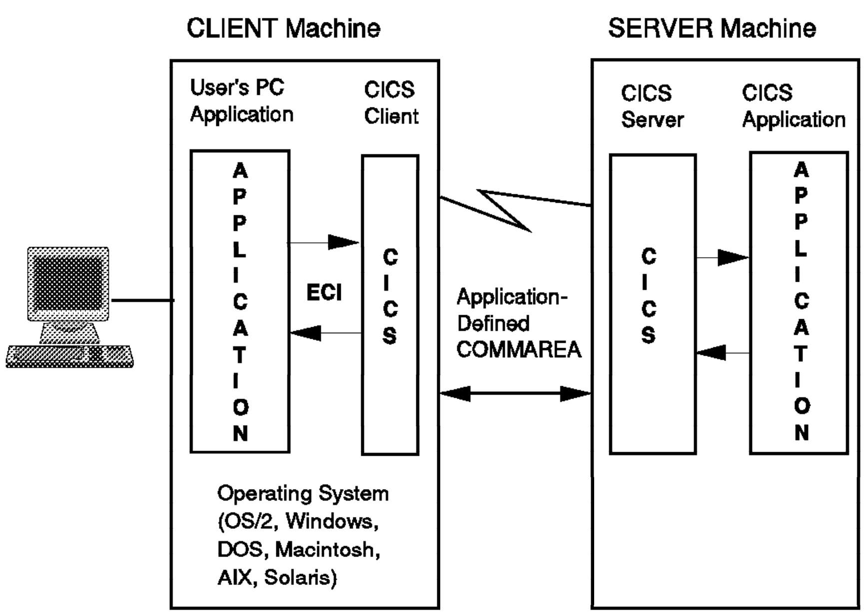

Figure 15. CICS ECI Model

For information about programming with the ECI, see *CICS Family: Client/Server Programming*, SC33-1435.

4.2.2 CICS External Presentation Interface

With the CICS EPI, your existing CICS applications can exploit GUIs on workstation-based CICS clients without having to be changed.

The EPI allows a GUI front end to be added to an existing CICS application without changing the CICS application. The CICS server application still sends and receives 3270 data streams to and from the client application as though it were conversing with a 3270 terminal. The client application captures this data and processes it as desired, usually displaying the data with a non-3270 presentation product such as a GUI.

Applications written with a GUI front end have proven to be easy for end users to work with and often simplify the task of learning a new application because the interface is the same no matter which facilities the application uses.

The limit of 15 concurrent EPI calls has been removed for the CICS Client for OS/2 Version 1.1. For all other CICS Clients there may be up to 15 concurrent EPI calls.

Figure 16 shows the CICS EPI model.

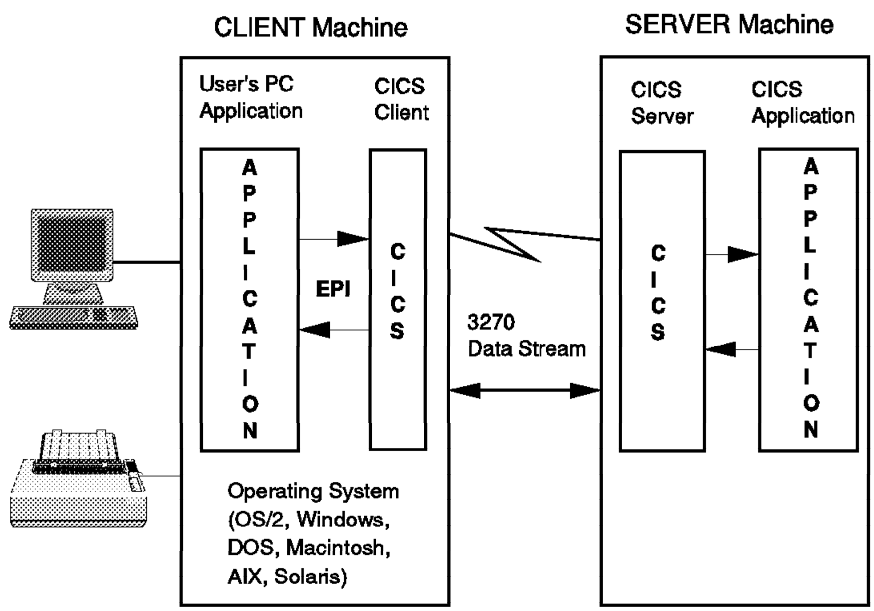

Figure 16. CICS EPI Model

For information about programming with the EPI, see *CICS Family: Client/Server Programming*.

4.2.3 CICS 3270 Terminal Emulator

CICS 3270 terminal emulator support enables a client workstation to function as a 3270 display for CICS applications, without the requirement for a separate 3270 emulator product. This support provides flexible client workstation capabilities without the need for extensive software outlay. Because each client (except CICS Client for DOS) can run multiple CICS 3270 emulation sessions, the hardware required is reduced, and end users can see multiple 3270 emulator sessions from one or more servers, all on one workstation.

By means of mapping files, you can customize the client emulator′s screen color attributes and keyboard settings. Thus your users can tailor their workstations to their own preferences or comply with company-standard keyboard layouts.

For the CICS for AIX servers (see Table 4 on page 57) the terminals are autoinstalled; that is, you do not have to predefine the terminals to the CICS server.

You can start the CICS 3270 emulator session by issuing the *cicsterm* command at the command input line. cicsterm is a 3270 terminal emulator that is designed specifically to work well with CICS. cicsterm performs terminal emulation and is responsible for displaying the 3270 data on the screen.

The CICS Client for AIX and CICS Client for Sun Systems provide another facility called *cicsteld* that allows connections to a CICS for AIX region using a generic 3270 Telnet client. You can use the particular 3270 terminal emulator of your choice at the client machine. This 3270 terminal emulator works as a Telnet client. It receives your input (through the keyboard) and communicates with cicsteld, which acts as a Telnet server on a destination machine. Then cicsteld communicates with the CICS for AIX region. cicsteld performs no 3270 emulation. It converts data into the 3270 Telnet protocol (EBCDIC 3270 data) and sends it to the Telnet client for display.

Figure 17 shows how a CICS for AIX region is accessed by a variety of clients. The CICS Telnet server (cicsteld) can reside in the same host as the CICS for AIX region.

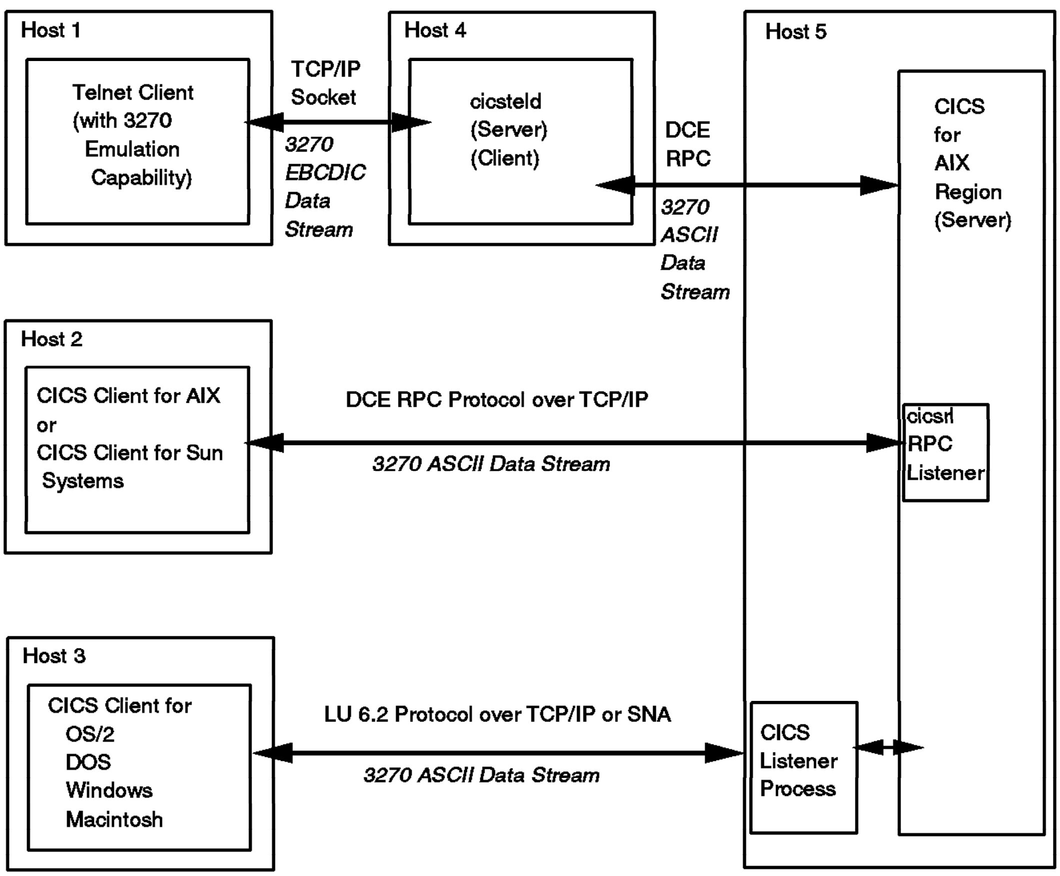

Figure 17. Relationship between Terminal Users and CICS for AIX Regions

4.2.4 CICS 3270 Client Printer Support

Client printer support is the ability to define a printer terminal on the client workstation. This support enables CICS applications running on the server to direct output to the client-attached printer. CICS 3270 client printer support uses CICS 3270 emulation functions.

You can direct the output to a physical printer attached, for example, to the LPT1 port, or you can specify a command to process the data into a format suitable for special-purpose printers.

With the CICS Client for AIX or CICS Client for Sun Systems, you can start the CICS 3270 Client printer support by using the *cicstermp* command. The cicstermp command performs a fixed-size printer emulation of 64 lines by 132 columns and ignores print attributes such as highlight, underline, high intensity, and color.

The CICS 3270 Client printer support is invoked by the *cicsprnt* command when you use CICS Client for OS/2, CICS Client for DOS, CICS Client for Windows, or CICS Client for Macintosh.

4.2.5 Benefits of Using CICS Clients

CICS Clients provide you with many benefits. Here are some of the main benefits:

Easy migration CICS Clients enable existing CICS users to move easily and inexpensively into client-server computing in a low-risk way.

Inexpensive hardware You can use existing low-function workstation hardware for your CICS Clients to gain access to CICS applications and data on the CICS for AIX servers.

Protected investment CICS Clients protect your investment in existing CICS applications designed for use with 3270 devices by enabling the EPI to allow these applications to interface with workstation facilities and productivity aids.

Exploit GUIs With the use of CICS Clients, your CICS applications can be optimized by locating the business logic on the server and the presentation logic on the client. Thus the usability and productivity of GUIs are exploited.

Concurrent server access A CICS Client can concurrently access multiple CICS applications on multiple CICS for AIX servers. Because client applications can access applications and databases on many servers at the same time, their power is considerably enhanced. CICS users on the supported platform can write new applications that use the client-server application server model.

Interface to nonterminal devices CICS Clients with the ECI offer you a convenient method of interfacing a nonterminal device, for example, bar-code readers and image scanners, with CICS for AIX servers.

Automatic data exchange CICS Clients provide a mechanism to enable automatic data exchange between desktop applications (for example, a spreadsheet) and data in a CICS for AIX server.

Platform advantages Workstation clients provide you with the advantages of the platform on which they run, that is, development tools, modern workstation interfaces (GUI, multimedia), and low cost.

Autoinstall CICS for AIX servers can autoinstall client connections and terminals. This capability greatly eases your systems management.

Novell networks In addition to IBM networks, you can use networks running Novell Netware for APPC-attached clients.

Local printers You can use local printers attached to desktop workstations to print output sent from CICS for AIX server applications.

Security CICS Clients provide userid and password security features.

NLS With national language support (NLS), your end users can receive CICS Client messages in their native language. The national languages currently supported by both CICS for AIX and CICS Clients are U.S. English, French, German, Italian, Spanish, Japanese, Korean, and simplified Chinese. (CICS Client for Macintosh does not support Japanese, Korean, or simplified Chinese.)

Workstation customization You can customize the colors of your CICS Client terminal emulator and remap your keyboard layout.

Distribution of clients On a distributed network you can install multiple copies of the CICS Client for OS/2 and CICS Client for Windows onto many workstations using a distributed management product, for example, IBM′s NetView Distribution Manager (NetView DM).

4.3 Versions

IBM Transaction Server for AIX provides a number of CICS Clients that can connect to a CICS for AIX server. They are:

- CICS Client for AIX, Version 2.1.1
- CICS Client for Sun Systems, Version 2.1.1
- CICS Client for OS/2, Version 1.1
- CICS Client for DOS, Version 1.1
- CICS Client for Windows, Version 1.1
- CICS Client for Macintosh, Version 1.1

These clients cover a number of different hardware platforms: the CICS Client for AIX and CICS Client for Sun Systems versions are for the appropriate UNIX-based operating systems; CICS Client for OS/2, CICS Client for DOS, and

CICS Client for Windows are for Intel-processor-based platforms; and CICS Client for Mactintosh is for the Motorola processor and Power Macintosh platforms.

CICS Client for OS/2, CICS Client for DOS, CICS Client for Windows, and CICS Client for Macintosh together constitute the *IBM CICS Clients Version 1.1* family. They are capable of communicating with other CICS server products within the CICS family. However, for our discussion, the CICS server is assumed to be a CICS for AIX Version 2.1.1 system.

CICS Client for AIX and CICS Client for Sun Systems can communicate with any of the CICS on Open System servers. Again for our discussion, the CICS server is assumed to be a CICS for AIX Version 2.1.1 system.

CICS Clients support two communication protocols:

TCP/IP Transmission Control Protocol/Internet Protocol originated from the UNIX operating system but is now supported by many operating systems including AIX, OS/2, VM, MVS, DOS, and Apple Macintosh.

APPC Advanced-Program-to-Program Communication is a commonly used term for the verbs and services included in the SNA LU 6.2. An LU 6.2 communication session provides device-independent application-to-application communication.

Note: A third communication protocol, Network Basic Input/Output System (NetBIOS), is supported by some CICS Clients. NetBIOS is a communication interface for the IBM LAN and Novell NetWare LAN environments. AIX does not support NetBIOS, and hence it is not relevant to our discussion.

The protocols and functions supported for each of the CICS Clients varies. Table 4 provides the details by client version of the protocols and functions that are supported. For each client, there may be support for more than one protocol and function. You can therefore use the protocols and functions that are best suited to your network environment.

Table 4. Protocols and Functions between CICS Clients and CICS for AIX Server

Server	**Client**	**TCP/IP**	**APPC**	**ECI**	**EPI**	**Auto-install**	**cicsteld**
CICS For AIX V2.1.1	DOS	☺	☺	☺	☺	☺	–
	Windows	☺	☺	☺	☺	☺	–
	OS/2	☺	☺	☺	☺	☺	–
	Macintosh	☺	☺	☺	☺	☺	–
	Sun	☺	–	☺	☺	☺	☺
	AIX	☺	–	☺	☺	☺	☺

Key:

☺ = supported
– = not supported

Notes:

- EPI always incorporates CICS 3270 terminal emulation and CICS 3270 Client printer support.
- With autoinstall, you do not have to predefine the CICS Client to the CICS server; that is, control table definitions are automatically created for the client at the CICS server.
- The CICS Telnet server (cicsteld) enables Telnet clients capable of emulating an IBM 3270 Information Display System to connect to a working CICS region.

4.4 Sample Network Configurations

CICS Clients can access CICS for AIX servers in your network in numerous ways. In this section we discuss some of these network configurations.

Two communication protocols, APPC and TCP/IP, are applicable for connections with the CICS for AIX servers. If you are already using a particular protocol, you may want to continue with that protocol alone.

The Table 3 on page 42 shows the four communication methods available for connections to the CICS for AIX server. As none of the CICS Clients supports communication at synchronization level 2, the Encina PPC TCP/IP and Encina PPC Gateway methods are irrelevant in our discussion. The CICS family TCP/IP and local SNA methods are applicable for the connections between CICS Clients and the CICS for AIX server.

If you have to connect different types of networks, for example, token-ring and Ethernet networks, you probably want to use TCP/IP. Indeed, TCP/IP is assuming ever greater importance and use in Internet communication. Also TCP/IP is the default communication protocol with the AIX operating system. You do not have to install an additional product as is the case with APPC.

4.4.1 TCP/IP

Using TCP/IP, all clients on a LAN can connect to the CICS for AIX server. Figure 18 shows CICS Clients communicating with a CICS for AIX server through TCP/IP.

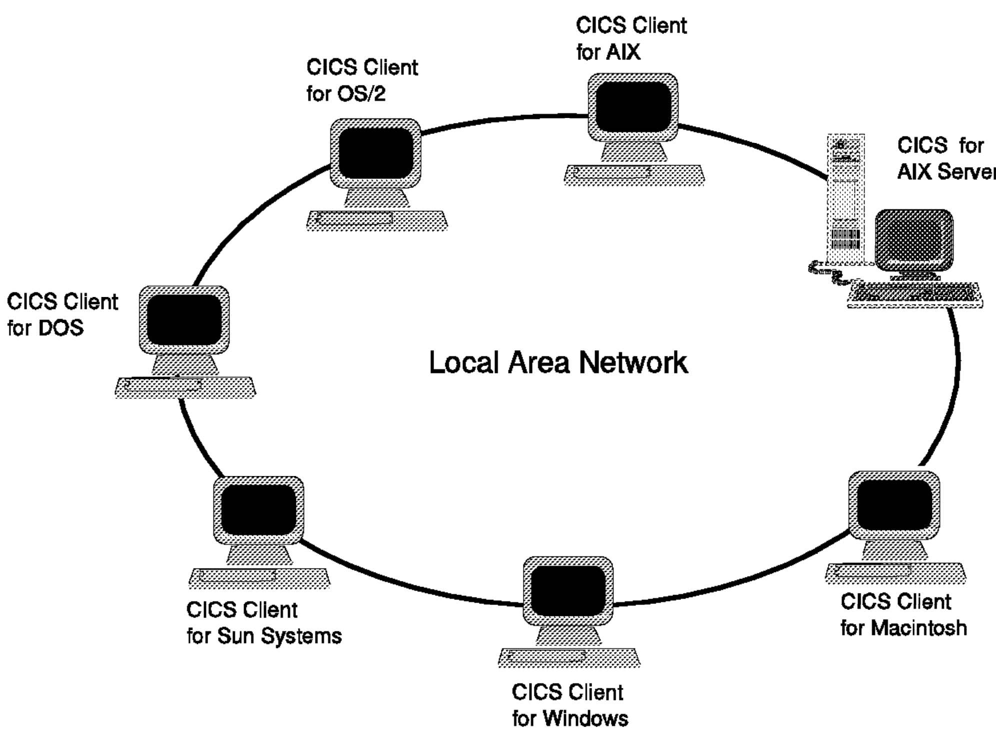

Figure 18. CICS Clients on a LAN Communicating over TCP/IP

4.4.2 APPC

Using APPC, CICS Client for OS/2, CICS Client for Windows, CICS Client for DOS, and CICS Client for Macintosh can connect to the CICS for AIX server. This communication is commonly achieved through a LAN and a workstation acting as a SNA gateway. However, for the CICS Client for OS/2 and CICS Client for Windows it is also possible to communicate directly with a CICS for AIX server without the need for a gateway workstation (see Figure 19). Examples include CM/2 for the CICS Client for OS/2, and IBM APPC Networking Services for Windows for the CICS Client for Windows.

A supported SNA product, for example, SNA Server for AIX, must be installed on the same machine as the CICS for AIX region if local SNA support (see Table 3 on page 42) is used.

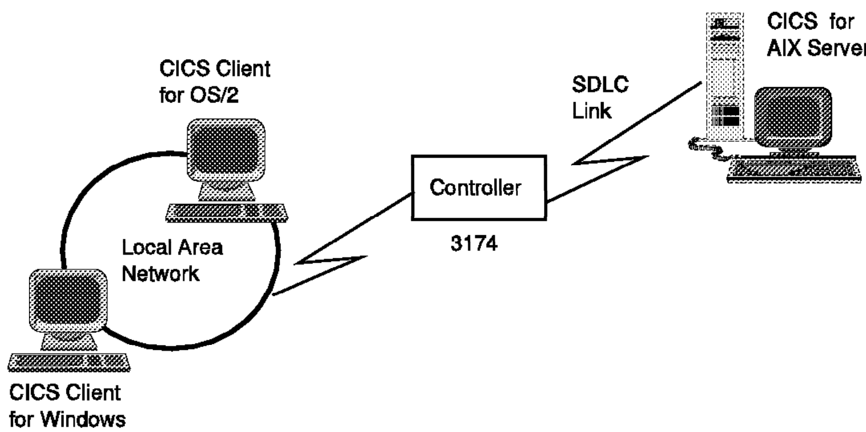

Figure 19. CICS Clients Communicating over APPC

For scenarios using APPC through a LAN and a SNA gateway, see *CICS Clients Unmasked*, SG24-2534-01.

4.5 Planning Considerations for CICS Clients

In deciding which CICS Client to use, you may believe that the answer is predetermined because your decision largely depends on the existing hardware and operating system on the desktop of the end user. This probably holds true in some cases, but not in all. The assumption that the end user does not have the budget to acquire a newer, better desktop computer and possibly a different operating system is not always correct.

The connection to the CICS for AIX server from the desktop machine through the appropriate CICS Clients enables the end user to run the mission-critical OLTP applications while able to operating existing workstation-based applications. You, as a planner. have to review the functions provided by each type of CICS Clients, cross-check them with the needs of the proposed CICS system or application, and then select the best CICS Client. Getting the functions that are needed for users to do their jobs correctly and efficiently will save you more in the long term than simply not paying for the upgrade of the desktop machines now. Higher efficiency and productivity gains cover the outlay of the capital investment in most cases. For example, a CICS Client for AIX ECI application is the only CICS Client that CICS Systems Manager for AIX supports for workload balancing. The Workload Client code integrates and works well with CICS Client for AIX. If you want to fully exploit the workload balancing capability of the CICS Systems Manager for AIX in association with your CICS for AIX servers, you have to upgrade the bulk of the workstations (if not all) to RISC System/6000 workstations or equivalents so that you can deploy the CICS Client for AIX on them. The end result is that none of your CICS for AIX servers is overloaded, and good consistent response time is achieved. Thus users are happy because they experience more even response times, and availability can be greater if you configured your CICS for AIX regions suitably. This in turn can lead to improved public image of your enterprise and greater efficiency from the staff who work in it.

In the sections that follow we cover the requirements for each CICS Clients, stating hardware and software requirements, limitations, and anything that you should be aware for a smooth implementation. All of these considerations are based on the assumption that the CICS server is a CICS for AIX region.

4.5.1 CICS Client for OS/2

CICS Client for OS/2 requires a machine that supports IBM OS/2 Version 2.0 or later. This implies a machine with an Intel 386SX or higher processor.

The appropriate network adapter cards and/or gateway cards are required for connection to the enterprise network. These cards enable the connectivity between the client machine and the CICS for AIX server machine.

The disk space required for the client code of CICS Client for OS/2 is about 1 MB. Additional disk space is required if you select the samples and documentation installation options. There is also an additional requirement if you select the CICS gateway for Lotus Notes and CICS Internet gateway components of CICS Client for OS/2.

To benefit from NLS support when communicating with a CICS for AIX server, CICS Client for OS/2 can run under any of the following operating systems:

- IBM OS/2 Version 2.0 or later
- IBM OS/2 J (Japanese) Version 2.0 or later
- IBM OS/2 H (Korean) Version 2.0 or later
- IBM OS/2 P (Simplified Chinese) Version 2.0 or later
- IBM OS/2 Version 3.0 (Warp)

CICS Client for OS/2 requires a product providing TCP/IP or APPC support. For example, OS/2 Warp Connect provides TCP/IP support and IBM Communications Server provides APPC support.

CICS Client for OS/2 provides its own LU 6.2 emulation to connect to a CICS for AIX region over TCP/IP and SNA networks. CICS for AIX supports requests from the CICS Client for OS/2 by means of a *listener process*. This multithreaded process (called *cicsip* in the case of TCP/IP, and *cicssl* in the case of SNA) communicates with the client process on the client machine. The communication between these processes imitates the flows between CICS regions. The client process, therefore, appears to the CICS for AIX server as a limited terminal owning region (TOR).

As the data flows are the same as those between CICS systems, the CICS for AIX server requires a communications definition (CD) entry for the CICS Client for OS/2 system. When the client process is started on the client, flows that start the initialization transaction (CCIN) are sent to the server. This transaction is responsible for performing an autoinstall of the appropriate CD entry for the CICS Client for OS/2 system. After the CCIN has run, all further requests are the ECI, EPI, or cicsterm request, depending on the type of client.

4.5.1.1 TCP/IP Connection

The CICS for AIX region must have a listener definition (LD) entry defined as an additional method to listen for incoming requests from the CICS Client for OS/2 across the TCP/IP network. The Protocol attribute in the LD entry should be set to TCP. The cicsip listener process is started at region startup. It listens on a TCP/IP port. The standard default port number is 1435. If you choose to use a nonstandard port, you must add an entry to the /etc/services file.

The CD model used by CCIN for autoinstall must have the ConnectionType attribute set to cics_tcp for the CICS family TCP/IP support.

The optimum number of connections simultaneously active through a single port is about 200, and the maximum is about 450 connections. Each time a CICS Client for OS/2 connects to the CICS for AIX server, a thread is started in cicsip and a connection is established. If you need more than 450 connections, there are two choices:

- Add a new adapter card to the machine

  This increases the adapter bandwidth. The port address is the same for all of the client systems, but the server Internet address is different. Two LDs are configured—one to listen exclusively on one network adapter and one to listen on the other. The TCPAddress attribute in the LD is used to specify the server Internet address.
- Use a second port

  The Internet address of the CICS for AIX server machine remains the same, but the port address is different. The TCPAddress attribute in the LD is left blank, and port addresses are explicitly defined.

If your CICS for AIX server machine manages many TCP/IP connections, it may run out of space in the mbuf pool (an area of memory allocated by TCP/IP when AIX is initialized). You can check the size by noting the value of *thewall* in the output of the *no -a* command. The default size of thewall is 2 MB, which supports up to about 800 CICS family TCP/IP connections. If you have to support more than 800 connections, you have to increase the value of thewall accordingly.

4.5.1.2 SNA Connection

The CICS for AIX region must have an LD entry defined as an additional method to listen for incoming requests from the CICS Client for OS/2 across the SNA network. The Protocol attribute in the LD entry should be set to SNA. The cicssl listener process is started at region startup. The CD model used by CCIN for autoinstalling the client must have the ConnectionType attribute set to local_sna.

The initial planning effort lies in the tailoring of the supported SNA product that provides the underlying SNA services to be used by the CICS for AIX region. IBM SNA Server for AIX Version 3.1 is an example of a supported SNA product. For more detail refer to *CICS on Open Systems Intercommunication Guide*, SC33-1564-01, and *SNA Server for AIX: Planning and Performance Guide*, SC31-8220.

4.5.1.3 Autoinstall User Exits

The CCIN calls user exit DFHCCINX when it is about to autoinstall the CD entry for the CICS Client for OS/2 system. If your own version of this user exit is in effect, review its logic to ensure that the CD attributes to be altered are compatible with those of the model CD entry.

When the first EPI or cicsterm request arrives at the CICS for AIX server the CCIN has run, the CTIN transaction carries out the autoinstall of a workstation definition (WD) resulting in a four-character terminal identifier of the format =xxx, where xxx is a character string chosen by CICS; the first character of = is fixed. If your autoinstall user exit programs do not expect the = character, you may have to review their logic to prevent potential problems.

4.5.1.4 Security

The userid and password are the main security mechanisms with the CICS Client for OS/2 system. The CICS Client for OS/2 holds only a single userid and password for each CICS for AIX server that it connects. You specify the userid and password for the particular CICS for AIX server either in the client initialization file (default name CICSCLI.INI) or by using the *cicscli* command. Also you must ensure that a user definition (UD) entry with matching userid exists in the connecting CICS for AIX region. Otherwise the default userid of the CICS region is used. In the case of cicsterm, you can enforce the initial transaction to be invoked to be CESN, the CICS-supplied sign-on transaction. This would cause the user to authenticate again.

4.5.1.5 3270 Keyboard Mapping

The keyboard mapping for 3270 emulator operation is defined in a keyboard mapping file. CICS Client for OS/2 supplies a default file, CICSKEY.INI, in the \CICSCLI\BIN directory. We recommend that you create your own customized mapping file. To identify this keyboard mapping file, you either specify the /k option on the *cicsterm* command or specify the CICSKEY environment variable in the CONFIG.SYS file.

4.5.2 CICS Client for DOS

CICS Client for DOS requires an IBM-compatible PC that supports the DOS operating system and has sufficient memory for the required software components.

The disk space required for the client code of CICS Client for DOS is about 0.9 MB. Additional disk space is required if you select the samples and documentation installation options.

To benefit from NLS support when communicating with a CICS for AIX server, CICS Client for Windows can run under any of the following operating systems:

- Microsoft or IBM PC DOS 3.3, DOS 4.0, DOS 5.0, or later
- Windows J (Japanese) Version 4.0 or later, Version 5.0/V or later
- Windows H (Korean) Version 4.0 or later, Version 5.0/V or later

A CICS Client for DOS can also run in the *DOS box* of OS/2.

CICS Client for DOS requires a product providing TCP/IP or APPC support. For example, IBM TCP/IP for DOS Version 2.1.1 or later provides the TCP/IP support. The particular product providing APPC support depends on the actual network configuration. For more detail see *CICS Clients Administration Version 1.1*, SC33-1436-01.

All of the configuration considerations for CICS Client for OS/2 are applicable to CICS Client for DOS. Please review 4.5.1, “CICS Client for OS/2” on page 60, replacing CICS Client for OS/2 with CICS Client for DOS and substituting AUTOEXEC.BAT for CONFIG.SYS.

4.5.3 CICS Client for Windows

CICS Client for Windows requires an IBM-compatible PC that supports the Windows operating system and has sufficient memory for the required software components. A minimum of 2 MB of memory is recommended.

The disk space required for the client code of CICS Client for Windows is about 0.9 MB. Additional disk space is required if you select the samples and documentation installation options.

To benefit from the NLS when communicating with a CICS for AIX server, CICS Client for Windows can run under any of the following operating systems:

- Windows Version 3.1 or later
- Windows J (Japanese) Version 3.1 or later
- Windows H (Korean) Version 3.1 or later
- Windows for Workgroups Version 3.1.1

CICS Client for Windows requires a product providing TCP/IP or APPC support. For example, IBM TCP/IP for DOS Version 2.1.1 or later provides the TCP/IP support, and IBM APPC Networking Services for Windows Version 1.0 provides APPC support.

All of the configuration considerations for CICS Client for OS/2 are applicable to CICS Client for Windows. Please review 4.5.1, “CICS Client for OS/2” on page 60 section, replacing CICS Client for OS/2 with CICS Client for Windows and substituting AUTOEXEC.BAT for CONFIG.SYS.

4.5.4 CICS Client for Macintosh

CICS Client for Macintosh requires any machine containing the Motorola M68020 (or higher) processor, for example, an Apple Classic II, or an Apple LC II. The PowerMac is also supported. For CICS Client for Macintosh, a minimum of 4 MB of memory is recommended. However, if other software components are to run concurrently with the CICS client, the total memory requirement may be greater.

CICS Client for Macintosh is compatible with the Virtual Memory mode of System 7 (accessible from the Memory Control Panel), although use of this Macintosh feature may affect performance.

The disk space required for the client code of CICS Client for Macintosh is about 1 MB. Additional disk space is required if you select the samples and documentation installation options.

NLS support is not available when using CICS Client for Macintosh to connect to a CICS for AIX server as the Macintosh operating system does not support NLS.

CICS Client for Macintosh can run under Macintosh System 7.1 or later. Apple Shared Library Manager (ASLM) Version 1.1.1 or later is also required.

CICS Client for Macintosh requires a product providing TCP/IP or APPC support. For example, MacTCP Version 2 or later can be provided by the Apple TCP/IP Connection for Macintosh product, and Apple SNA-ps 3270 Gateway Client is one way to provide APPC support.

The majority of the configuration considerations for CICS Client for OS/2 are applicable to CICS Client for Macintosh. Please review 4.5.1, "CICS Client for OS/2" on page 60 section, replacing CICS Client for OS/2 with CICS Client for Macintosh. There is a minor difference in the way in which CICS Client for OS/2 and CICS Client for Macintosh identify the keyboard mapping file. CICS Client for Macintosh identifies the keyboard mapping file through the creation of an appropriate options file for the cicsterm command. This is due to a difference in the operation of CICS Client for Macintosh. A CICS Client for Macintosh Administration Utility provides the same functions as the *cicscli* command supplied by CICS Client for OS/2, CICS Client for Windows, and CICS Client for DOS. The plain-text ASCII files containing options (called *options files*) are used to specify the command options.

4.5.5 CICS Client for AIX

CICS Client for AIX Version 2.1.1 runs on the RISC System/6000 machine with the AIX Version 4.1 operating system. The communication protocol used has to be TCP/IP.

CICS Client for AIX uses DCE RPCs to communicate with the CICS for AIX region. An incoming RPC from a client is processed on one of the threads of the transaction scheduler. The default number of threads is 10 times the number of application servers that is specified in the Region Definition stanza, subject to a limit of the number of threads allowed per process by the operating system. The limit on the number of listener threads is approximately 500. The exact number will vary depending on the number of other threads that exist in the transaction scheduler. See 3.2.1.2, "Transaction Scheduler" on page 29 for more details of scheduling activity. You specify the number of RPC listener threads through the RPCListenerThreads attribute in the region definition (RD).

In an environment in which DCE CDS and security are used, you have to enable DCE access on each machine that is running CICS Client for AIX. CICS for AIX supplies the *cicssetupclients* command to assist you in doing this. DCE base services must be configured for the client machines to have the DCE RPC capability. This configuration must be done regardless of the physical location of the client or whether the DCE CDS is being used.

4.5.5.1 User Access

You need to understand how your users access the CICS for AIX server with the CICS Client for AIX. There are a number of ways of accessing CICS:

- ECI

  The ECI call is an indirect way of accessing CICS resources in the CICS for AIX region. It behaves like a DPL request. You have to invest programming effort in workstation applications in order to issue the ECI calls. End users do not need any knowledge of how to access the CICS for AIX region. You have to consider the required programming effort against the benefits of using the ECI.

- EPI

  The EPI enables a non-CICS application program, running in the client machine, to access the CICS for AIX region as if it was from a 3270 terminal. As the EPI uses DCE RPC, DCE Base Services are required on the client machine. With EPI you can have up to 15 concurrent logical terminals on one client workstation. However, investment in programming the non-CICS application is required to exploit the ability to serve several logical terminals from a single EPI session.

  With EPI, security is handled by DCE and can be controlled in the same way as for cicsterm.

  The EPI uses multiple threads when processing data to and from the CICS for AIX region. cicsterm, however, communicates synchronously with aixterm. Therefore, we believe that the EPI is more efficient than cicsterm or cicsteld even after adding the overhead of a display manager such as X Windows or Motif. The EPI provides a good opportunity to add a GUI front end to existing applications. You have to consider the required programming investment against the efficiency gain when using the EPI.
- cicsterm

  cicsterm provides 3270 emulation for AIX terminals and is probably the most straightforward way of accessing the CICS for AIX region. It runs as an AIX process, on the server machine or on the desktop client machine. Running cicsterm on the CICS for AIX server machine uses up to 40% of the server′s processing and hence greatly undermines the server′s performance.

  If DCE CDS is used, for successful connection the CICS Client for AIX client machine must belong to the same DCE cell as the CICS for AIX server machine. If DCE CDS is not in use, see 5.2.2, "Configuration without DCE CDS and Security Services" on page 73.

  For security, you can define the level of authentication used for RPCs between the CICS for AIX region and the cicsterm. The higher the authentication level used, for example, pkt_integ instead of none, the greater the impact on performance. Therefore you have to balance between the security requirements and an acceptable level of performance. To protect from possible eavesdropping, you can encrypt the information flow across the connection.

  Each cicsterm process requires about 560 KB of real storage for its working set and about 1.4 MB of virtual storage. Using these figures you can estimate the number of cicsterm processes that a particular RISC System/6000 machine can support or you can calculate the storage requirements for a given number of cicsterm processes.
- cicsteld

  CICS Client for AIX provides support for Telnet 3270 clients through the cicsteld process. The cicsteld process that acts as a 3270 Telnet server has to run on a machine located in the same DCE cell as the CICS for AIX server machine. The Telnet 3270 client does not use DCE and hence is not required to be in the same DCE cell as the server machine. A limitation of this method is that you cannot encrypt, nor can you protect against eavesdropping between the Telnet 3270 client and the cicsteld process.

  cicsteld uses less processor power than cicsterm because the keystroke handling and 3270 emulation costs are incurred by the machine running the Telnet 3270 client. You can therefore potentially run the cicsteld process in the same machine as the CICS for AIX region. However, for maximum throughput, we recommend that you not run the cicsteld processes on the same machine as the CICS for AIX region. The 3270 Telnet client consumes much of the processor time on the client machine, so using a fast client machine would improve the response time of the CICS transaction.

The 3270 Telnet client program, not cicsteld, provides the 3270 emulation. You must ensure that the 3270 Telnet client program used provides the level of data stream support, particularly the attributes settable by CICS basic mapping support (BMS), required by your CICS transactions.

The cicsteld command authenticates to DCE with a DCE principal and password from a keytab file when it is started up. The subsequent DCE RPC call by the cicsteld is considered as an authenticated request when a 3270 Telnet client requests cicsteld to connect to the CICS for AIX region. The 3270 Telnet client is not asked to provide a password. The significance of this is that access to CICS transactions and resources on your CICS for AIX region is granted to any Telnet client user who is aware of the port that cicsteld is listening on. There is a possible security exposure for your transactions and resources if you choose an inappropriate DCE principal. Our recommendations for creating a secure environment when using cicsteld are:

- Restrict execution access to the cicsteld program to a controlled set of users
- Set up cicsteld to run with the DCE principal associated with the CICS for AIX region′s default userid that is the default for unauthenicated users
- Encourage Telnet clients to sign on to CICS by ensuring that cicsteld invokes the CESN transaction as the first transaction.

For planning purposes, the storage requirements of the cicsteld process are close to that of the cicsterm process. An instance of cicsteld must be started for every connection between a 3270 Telnet client and a CICS for AIX region. Therefore the number of running cicsteld processes reflects the number of connections from the 3270 Telnet clients. The increased level of authentication degrades performance.

4.5.5.2 Autoinstall User Exit

The CHAT transaction is responsible for performing an autoinstall of the WD for the CICS Client for AIX, generating a terminal identifier based on the first two characters of the client system. If you have your own version of the autoinstall user exit in effect and the naming convention of the client system is to maintain the same first two characters for all client systems, you may have to review its logic to prevent possible duplication of terminal identifiers.

4.5.5.3 3270 Keyboard Mapping

The keyboard mapping for 3270 emulator operation is defined in a keyboard mapping file. CICS for AIX provides a number of keymap files with a prefix name of 3270keys in the usr/lpp/cics/etc directory for a variety of terminals such as 3151, hft, vt100, vt200, xterm, and sun-cmd. You may want to use the 3270 key mappings provided with CICS instead of the defaults provided with the AIX operating system. In this case, you have to link or copy the target keymap file to the appropriate directory to make it effective. Alternatively you could make your own copy of the keymap file. It is important to have the correct 3270 keyboard mapping in place for each CICS end user. Otherwise user frustration due to the wrong mapping of keys negates the benefits of using CICS Client for AIX

4.5.5.4 Workload Management

CICS Client for AIX is the only CICS Client supported by CICS Systems Manager for AIX for workload management. If you want to exploit the benefits of workload balancing, you have to use CICS Client for AIX to connect to the CICS for AIX servers and have the workload management infrastructure of CICS Systems Manager for AIX operational. For details about workload management, please read Chapter 8, "Using CICS System Manager for AIX" on page 111.

4.5.6 CICS Client for Sun Systems

CICS Client for Sun Systems Version 2.1.1 includes cicsterm, cicstermp, cicsteld, the EPI, and the ECI. These enable you to work with any CICS for AIX region in the same DCE cell as your Sun workstation. The Sun workstation can be any uniprocessor Sun SPARCstation, or any binary compatible SPARCserver or SPARCclassic. Solaris operating system V2.2 or above, and Transarc DCE for the Solaris operating environment Version 1.0.2 (Base Services) are required for successful execution of CICS Client for Sun Systems.

The CICS Client for Sun Systems runs on the Sun workstation machine and the CICS for AIX server runs on an IBM RISC System/6000 system. CICS Client for Sun Systems has to run off the server machine. Both machines should belong to the same DCE cell. They communicate using DCE RPCs over a TCP/IP connection.

Most of the points that we discuss about CICS Client for AIX are also applicable to CICS Client for Sun Systems. Please refer to 4.5.5, "CICS Client for AIX" on page 64 for these points. The exceptions are:

- A minimum of 24 MB of random access memory (RAM) is required for the first instance of the CICS Client for Sun Systems on the Sun workstation. Each subsequent instance of the CICS Client for Sun Systems requires an additional 1.5 MB.
- CICS Systems Manager for AIX does not support CICS Client for Sun Systems for workload management.
- You must ensure that the terminal models appropriate for Sun terminal types have been added to your CICS region terminal definitions (known as WD). The terminal types can be sun, sun-cmd, or xterm.
- CICS Client for Sun Systems provides sample key mapping files suitable for use with cicsterm, including 3270keys.vt100, 3270keys.sun-cmd, and 3270keys.xterm.

4.5.7 Additional Information

To help evaluate the IBM CICS Clients for your own needs, see:

- *CICS Clients Administration*, SC33-1436. The book is shipped with the product, but you can also order it separately.

- *CICS Family: Client/Server Programming*, SC33-1435. This book accompanies all CICS server and client products.

- *CICS Interproduct Communication*, SC33-0824-02. This book describes communication among the CICS products.

At the time of writing this chapter, CICS Clients Version 2.0 was announced. It provides:

- New additional CICS Clients for Windows 95 and Windows NT

- Support for object-oriented programming
- Gateway functions for CICS access to Lotus Notes and the Internet
- Major enhancements to NLS for more than 20 languages
- New package on CD-ROM with all clients in one package

4.6 Summary

CICS Clients offer the end users of your enterprise the best of both worlds. They can continue to use their familiar workstation applications on their existing workstations, and using an appropriate CICS Client, they can also access and run OLTP applications in the CICS for AIX servers.

The six different types of CICS Clients provided by IBM Transaction Server for AIX should meet your needs in most cases as they cover the most commonly used workstation operating system environments. The four common functions supported by all of the CICS Clients are the ECI, the EPI, CICS 3270 terminal emulator (cicsterm), and CICS 3270 client printer support (cicstermp or cicsprnt).

In this chapter we give you a conceptual overview of CICS Clients and the benefits that you can realize when using them to communicate with your CICS for AIX servers. Also we look at each type of CICS Client and discuss information that is important in the planning stages, covering issues such as capacity sizing, security, and performance implications.

Chapter 5. Interactions with DCE

To facilitate flexible and secure distributed processing across the operating systems of multiple vendors, a set of common services is required that will perform the following functions:

- Security

  It should be possible to authenticate users or processing elements to verify that they are who or what they claim to be.
- Dynamic location of processing components

  Components of a distributed processing system should have the flexibility of being able to be started on any accessible system without having to go around and register their addresses with each of the other components each time they are moved.
- Intercomponent communication

  Each of the components of a distributed application should have some means of communicating with the other components.
- Time synchronization

  With multiple distinct processors involved in supporting the components of a distributed application, some form of time synchronization service is required. Otherwise the clocks of the involved processors may fall out of synchronization, which can lead to complications should transactions have to be backed out, for example.

The OSF DCE is a set of integrated services designed to support the development and use of distributed applications. The OSF is a host for industrywide open systems research and development. Users, software developers, and hardware vendors share resources and ideas with the goal of improving the scalability, probability, and interoperability of computer systems. OSF DCE is operating-system independent and provides a solution to the problems of sharing resources in a heterogeneous, networked environment.

DCE is a layer of services that allows distributed applications to communicate with a collection of computers, operating systems, and networks. This collection, when managed by a single set of DCE services, is referred to as a DCE *cell*.

5.1 Services

DCE provides the following services for use by distributed applications:

- DCE Threads
- DCE Remote Procedure Call (RPC)
- DCE Security Service
- DCE Directory Service
- DCE Distributed Time Service
- DCE Distributed File System (DFS)

IBM Transaction Server uses all of these services except DCE DFS. We now discuss these services in more detail.

5.1.1 DCE Threads

Threads support the creation, management, and synchronization of multiple, concurrent execution paths within a single process. Threads provide a great deal of flexibility to application developers in a variety of areas, such as parallel execution, task control, exploitation of multiprocessor machines, and faster task switching. Neverthless, threads introduce considerable, additional complexity.

The processes in a CICS for AIX region use DCE threads for their multithreading support.

5.1.2 Remote Procedure Call

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

The RPC application has two sides: the client side, which calls the remote procedure, and the server side, which executes the procedure in its own address space. Clients and servers can be on different computers linked by communications networks.

DCE RPC supports authenticated communications between clients and servers. Authenticated RPC enables you to place additional data protection and authentication on data transmitted between clients and servers. Several levels of security are provided. Authenticated RPC is provided by the RPC run-time facility and works with the authentication and authorization services provided by the DCE security service.

Another important service provided by DCE RPCs is *endpoint mapping*. When a server used by CICS initializes, it writes its interface name and server process address, the *endpoint*, into an *endpoint map* database. On each host with servers, an RPC daemon process (rpcd) uses the endpoint map database to help clients identify servers. Each endpoint map database contains information for all servers on that host.

RPCs are used for communication between CICS Client for AIX and a CICS for AIX region, a CICS for AIX region and a CICS for AIX region, and CICS for AIX and Encina. Although it is possible to run CICS for AIX without the use of DCE CDS and security, RPC is still required.

5.1.3 DCE Security Service

Most multiuser operating systems provide some method to verify the identity of a user (authentication) and to determine whether a user should be granted access to a resource (authorization). In a distributed environment, a way has to be provided to authenticate requests made across the network and to authorize access to the network′s resources. There must also be a mechanism to protect network communications from attack. The challenge in a distributed environment is to provide these services transparently to both users and programs. For example, a user should not have to authenticate to each server in the network.

The DCE Security Service is made up of several parts.

- The Authentication Service allows processes on different machines to determine each other's identity (authenticate).
- The Privilege Service determines whether an authenticated user is authorized to access a server resource. The Privilege Service provides information that servers need to determine the access that should be granted to the user.
- The Registry Service manages a security database used to hold entries for all principals. A principal is a user, server, or computer that can communicate securely with another principal. The Registry Service is also used by administrators to maintain the list of principals known to DCE.
- The Audit Service detects and records security operations performed by DCE servers. This is new in OSF DCE 1.1.
- The Login Facility performs the initialization of the DCE environment for a user. It uses the Security Service to authenticate a user and returns credentials to the user. These credentials are then used to authenticate to other services in the DCE cell. The credentials expire after a set period of time or when the user exits from the DCE environment.

Most of these security components are transparent to the user.

The DCE security services can provide two services to CICS for AIX: authentication of users and authenticated RPCs. The use of authentication ensures that users who are requesting connection to a CICS for AIX region can be verified to be who they claim to be. The use of user authentication is a useful extension to CICS security. You should consider using this facility since CICS internal security relies on permitting access to users based on their userids. The more rigid and secure the authentication process is, the more effective your security policy can be. For details on CICS security see 3.1.1.2, "Security" on page 22. The use of authenticated RPCs can be used to protect access to the data that flows between CICS components. Several levels of authentication are available, from checking only at the beginning of an RPC to data encryption. Be aware that the higher the level of protection, the greater the processing overhead. If a protection level of *none* is considered as having no performance impact, the highest level, *packet privacy*, can be five times the impact of the *none* level.

5.1.4 DCE Directory Service

The DCE Directory Service provides a naming model throughout the distributed environment that enables users to identify, by name, network resources, such as servers, users, files, disks, or print queues. The DCE Directory Service includes:

- Cell Directory Service (CDS)
- Global Directory Service (GDS)
- Global Directory Agent (GDA)
- Application Programming Interface (API)

The CDS manages to locate names within the cell and is optimized for local access. It is a partitioned, distributed database service, and the partitions can be stored in different locations, thus allowing good scalability. The CDS can also be replicated, which affords good availability of the system. A cache mechanism improves the performance by reducing the number of times it has to be accessed.

The GDS is based on the CCITT X.500 name schema and provides the basis for a global namespace. The GDA is the CDS gateway to intercell communication. The GDA supports both X.500 and Internet addresses. If the address passed to the GDA is an X.500 address, the GDA contacts the GDS. If the address passed to the GDA is an Internet addresses, the GDA uses the Internet Domain Name Service (DNS) to locate the foreign cell. Both CDS and GDS use the X/Open Directory Service (XDS) API as a programming interface.

CICS for AIX uses the DCE CDS to:

- Help clients find servers

  The server could be a CICS for AIX region or an Encina server (PPC Gateway or SFS)
- Determine which port a server uses to listen for client requests
- Specify which protocol is used between the client and the server

CICS for AIX provides these facilities, so an enterprise could decide not to use DCE CDS.

5.1.5 DCE Distributed Time Service

DCE Distributed Time Service (DTS) provides time synchronization across distributed computers so that events occur in the network relative to the same universal time.

5.2 CICS for AIX and DCE

In 5.1, "Services" on page 69 we discuss the DCE services which CICS for AIX and Encina use. In this section we explain the differences between a configuration that uses DCE CDS and security services and a configuration that uses the CICS-provided facilities.

5.2.1 Configuration with DCE CDS and Security Services

Figure 20 on page 73 shows a configuration consisting of four systems: Host 1, Host 2, Host 3, and Host 4.

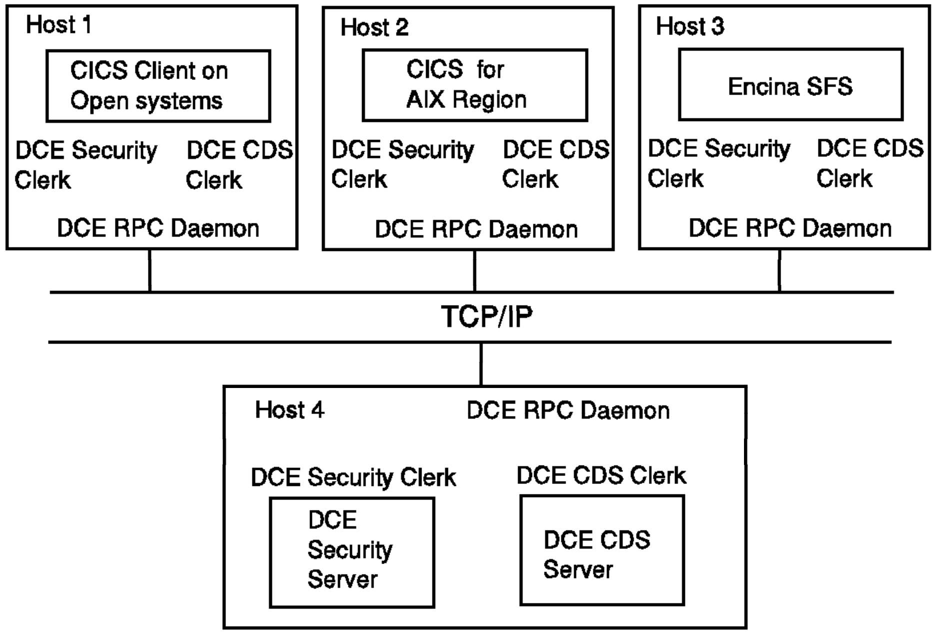

Figure 20. A Configuration with DCE CDS and Security Services

The CICS client, region, and Encina SFS that hold the CICS files and queues are each located on different systems. DCE CDS is used to help the client find the server. All systems are within the same DCE cell. Host 4 is configured as the DCE server for this cell. Host 1, Host 2, and Host 3 are configured as DCE clients. Each of the systems has a DCE CDS and security clerk. When a client needs to find a server, for example, the CICS client trying to locate the CICS region, a name-service lookup is made. This lookup may be satisfied by the information held by the local CDS clerk, or it may have to be passed through to the CDS server. If the request goes through to the CDS server, the information is subsequently held in the local CDS clerk cache of the requesting system for any future name-service lookups. This scheme minimizes the number of trips made to the DCE server to get information. The DCE RPC daemon is present on each system. RPC is used for intercomponent communication.

Each of the CICS components Figure 20 could be located on a single system if required. The use of DCE CDS allows the CICS components to be distributed easily and flexibly. A CICS for AIX region could easily be exported from one machine and then imported on another. When the CICS for AIX region is next started, its new location would automatically be determined, rather than through manual intervention.

In this configuration, users authenticate to the CICS for AIX region by using a DCE principal.

5.2.2 Configuration without DCE CDS and Security Services

When DCE CDS is not in use, all clients that have to contact a remote server must provide the host name where that remote server is located. This is done by setting the $CICS_HOSTS environment variable. The variable is set to a list of host names. The hosts named are all hosts with CICS regions to which CICS clients have to connect, Encina servers (PPC Gateway or SFS), and other CICS regions that the region needs to connect. For example, if $CICS_HOSTS=′hostA hostB hostC′, CICS will search hostA for string binding information for whatever region or server it is looking for. If the server or the region does not exist on that host, hostB is searched, and finally hostC if required. Optionally for CICS Client for AIX, the host name of the CICS for AIX region may be specified on the cicsterm command.

When any CICS region or Encina server is moved from one system to another, the $CICS_HOST variable must be updated on those systems which access that CICS region or Encina server. This scheme lacks the flexibility that DCE CDS has. Additionally if there are many host names specified in the $CICS_HOST variable, the search time to locate a CICS region or Encina server will be longer. Figure 21 shows a simple configuration consisting of two systems, host 1 and host 2. DCE CDS and security are not in use in this configuration. The DCE RPC daemon is still required on each system so that a client can call a server.

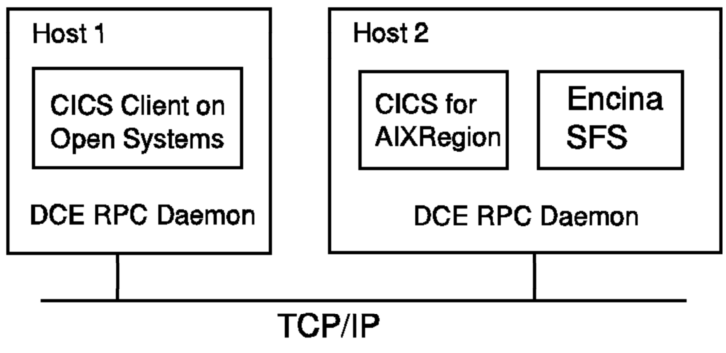

Figure 21. Configuration without DCE CDS and Security Services

As DCE is no longer in use there is no requirement for a DCE server and so there are no DCE clerks on each of the client nodes.

In this configuration users must authenticate themselves to a CICS for AIX region by using a CICS userid and password.

5.3 Planning Considerations for DCE

As the use of DCE CDS and security are optional, you must first decide whether to use them or not. We strongly recommend using DCE CDS and security for the following reasons:

- DCE provides a common, standards-based infrastructure across multiple vendors for distributed processing. Installations are typically multivendor. Software that is multivendor can simplify the process of administration and reduce the administrative overhead. It is possible to operate the mixed environment as a single DCE cell.
- DCE security provides comprehensive security facilities. The security facilities cover authentication, authorization, data integrity, and data encryption. CICS for AIX in comparison only provides limited authentication facilities. We recommend combining the DCE and CICS security methods.
- DCE CDS provides dynamic server location for requesting clients. In a complex environment where servers may move as configurations change or processors are upgraded, for example, the administrative overhead of doing server location is reduced. CICS for AIX does provide facilities for allowing clients to access to servers. Manual intervention is required, however, when a server is moved, to ensure that the clients can locate the repositioned server. This could become a significant overhead when a CICS for AIX

region is moved and all CICS Client for AIX machines from which the CICS for AIX region will be accessed need to have the $CICS_HOST variable updated. With DCE CDS this effort is spared and the clients automatically locate the CICS for AIX region.

If you decide to use DCE CDS and security services, you will have to consider the following:

- Decide how many DCE cells are to be configured.

  It is wise to minimize the number of DCE cells. You may decide to have one cell for production and another for test. We do not suggest having DCE cells by application, or CICS configuration. The fewer the better.
- Allocate a DCE cell server.

  Allocate a system that is to be configured as a DCE cell server. It should not be a heavily used system. Ensure that the chosen system has good connectivity to its client systems.
- Replicate the DCE cell server. For production systems consider replicating the server for increased availability. The loss of the DCE server can stop all work as users no longer become able to authenticate to DCE.
- CICS and Encina distribution

  Consider how you will allocate the different processing components over the available nodes. Although it is possible to have the CICS clients, CICS for AIX region, and Encina servers on different systems, it may not always be the most suitable choice. You may have to allocate the CICS for AIX region and Encina SFS on different systems so that the Encina SFS can be given more processing power, for example. With such allocations, however, it is not possible to use the performance optimization of FLT as the communication method between them.
- Minimize DCE reconfigurations

  Be aware of the consequences that switching DCE cells has on your CICS configuration. Before moving DCE cells, you will have to export any existing regions, unconfigure the system from its current DCE cell, reconfigure in the new DCE cell, and then import the region.

5.4 Summary

DCE provides a number of very important functions for a CICS for AIX system. These functions enable a client to discover the location of a server and provide varying levels of security, multithreading for the CICS for AIX run-time processes, and RPC, which is used in communication between CICS for AIX processes and an Encina SFS when present.

It is possible to develop a CICS for AIX configuration without installing DCE, however. The multithreading support used by the CICS for AIX processes and the RPC are provided though AIX. CICS for AIX does provide the necessary support to allow intersystem communication without the use of DCE CDS. CICS for AIX does provide security facilities. They do not allow for the same level of user authentication or RPC data encryption, however. Whether to configure DCE CDS and security services or not is a decision you must make on the basis of the requirements of your configuration. We recommend that you use DCE CDS and security services in any multisystem configuration.

Chapter 6. Interactions with Encina for AIX

Although DCE addresses many of the complexities of distributed computing, there are still additional requirements to facilitate the development of robust distributed applications. To the DCE-based environment, a set of transactional services must be added to provide data integrity in the event of concurrency and system failures. Transactional processing is fundamental to enterprise computing because it provides the necessary data integrity. An enterprise depends on the ability to maintain accurate and consistent information. Although error conditions may be rare, it is essential to cater for them. It is not realistic to simply rely on the law of probabilities that failures will not happen. Thus you must have software that understands the concepts of transaction-based processing, logging, and error recovery. CICS for AIX does not provide these facilities itself. It uses the facilities provided by the Encina family of products.

The Encina architecture is based on a two-tiered strategy:

- Expand the DCE foundation to include services that support distributed transaction processing and the management of recoverable data.
- Construct a family of high-function transaction processing services based on this expanded foundation.

The Encina family components of interest when discussing a CICS for AIX environment are:

- Encina Toolkit Executive
- Encina Toolkit Server Core
- Encina PPC Services
- Encina SFS

6.1 Services

Encina provides a number of services that are used by CICS for AIX:

- Transaction services, which ensure data integrity over distributed units of work and across the X/Open XA interface
- Communication facilities, which provide connection to an SNA network
- Data management facilities, to manage CICS queues and user files

The transaction services are essential to all CICS for AIX configurations. They provide the very core of transaction processing. The communication and data management facilities are optional, and you may use them or not depending on your requirements. Below we examine each of these services in detail.

6.1.1 Encina Toolkit Executive

Encina Toolkit Executive, also known as Encina Base, provides services that permit a node to initiate, participate in, and commit or back out distributed transactions. The services include transactional RPC, an extension of the DCE RPC that transparently ensures transactional integrity over distributed units of work.

Encina Base manages the two-phase commit process between a transaction manager (CICS) and a resource manager, such as a relational database or Encina′s SFS.

6.1.2 Encina Toolkit Server Core

Encina Toolkit Server Core, also known as Encina Server Core, is built on Encina Base to provide services for managing recoverable data, such as data that is accessed and updated by transactions. The services include an X/Open XA interface to permit the use of XA-compliant resource managers.

6.1.3 Encina PPC Services

Although distributed computing is playing an increasing role in enterprise computing, the majority of online data is still maintained on proprietary mainframes. For enterprises to develop suitable migration strategies, it is essential that new applications and systems can access this data with integrity. The mainframes largely use SNA-based networks. Most distributed computing takes place over TCP/IP. CICS applications must be able to communicate with other distributed systems using TCP/IP or mainframe-based systems using SNA. CICS for AIX in conjunction with Encina PPC services is able to provide one means of achieving this communication. See 3.3, “Intercommunication” on page 31 for alternative approaches.

PPC services are provided through the following components:

- Encina PPC Executive
- Encina PPC Gateway

6.1.3.1 Encina PPC Executive

Encina PPC Executive provides a mechanism for ensuring that transaction semantics are preserved across distributed processes. The PPC Executive is required for intersystem communication over a TCP/IP network and can be used for intersystem communication across an SNA network.

The PPC Executive contains support for LU 6.2 peer-to-peer communication, as well as an emulation of LU 6.2 over TCP/IP.

The PPC Executive can be run on any host that is executive enabled, provided that an Encina Toolkit Server Core is accessible within the environment. Thus Encina clients and servers running PPC Executive can carry on transactional LU 6.2 conversations, including two-phase commit control flows, over TCP/IP. The PPC Executive also serves as enabling technology for Encina PPC Gateway for AIX.

To understand how Encina PPC Executive is used in CICS for AIX intersystem communication over TCP/IP, see 3.3.2.2, “Encina PPC TCP/IP” on page 33.

6.1.3.2 Encina PPC Gateway

In conjunction with the Encina PPC Executive, the PPC Gateway Server provides transactional interoperability over an SNA protocol implementation of LU 6.2. Thus CICS applications can ship or accept transactions from other systems that use LU 6.2, such as IBM mainframe-based CICS, CICS for OS/2, and CICS/400.

To understand how Encina PPC Gateway is used in CICS for AIX intersystem communication providing two-phase commit support over SNA, see “Encina PPC Gateway” on page 35.

6.1.4 Encina SFS

The AIX File System and DCE Distributed File Service suffer from a number of limitations in their capacity to support TP applications. First, they do not offer transactional integrity or appropriate security features. More importantly, file I/O is restricted to flat byte streams. To represent record-oriented data, applications must impose structure on these streams, with the associated cost in complexity and performance. Finally, most AIX file systems are not log-based, so a restart after a crash involves the unacceptable cost of traversing entire disks.

Although RDBMSs address many of these issues, they, too, are inappropriate for many applications. In particular, RDBMSs typically hide from the application developer the details of how data is organized and laid out on disk. While often beneficial, real-time applications or those requiring the storage of very large amounts of data (many gigabytes) may demand greater control over data organization. Finally, there exists an extensive investment in APIs such as X/Open ISAM and VSAM, both in terms of programs and programmers.

The Encina SFS is a record-oriented file system, offering transactional integrity and supporting ISAM-compliant and VSAM-style APIs. SFS scales to support very large files spanning multiple disks and large numbers of concurrent users. For example, unlike many RDBMS designs, clients are not allocated their own SFS process, which substantially reduces context swap overhead. Instead, the SFS employs the Encina Toolkit Executive, the Encina Toolkit Server Core, and the AIX DCE services to support modern paradigms such as multithreaded access and nested transactions. Encina SFS also makes use of the underlying logging and recovery services to enable fast restarts and media recovery.

Using Encina SFS it is possible to manage the data associated with CICS user files, auxiliary temporary storage queues, intrapartition transient data queues, and locally queued automatic transaction initiation (ATI) requests. CICS applications access SFS files through the CICS API.

Although IBM mainframe-based CICS systems use VSAM for the file system, the differences between the SFS file system and VSAM are transparent to CICS application programs.

The Encina SFS is a server process that manages access to data stored in record-oriented files. An SFS is ideally suited for applications that manage large amounts of record-based data, for example, inventory records, customer orders, and employee files.

Storing data in SFS files provides the following advantages:

- Transaction protection.

  SFS provides both transactional and nontransactional access to data stored in an SFS file. The SFS uses the services of the Encina Toolkit Executive to recover from server problems, network outages, and media failures. With transactional access, the state of the SFS file after recovery reflects data changes from all committed transactions. Any transactions in progress at the time of the failure are either completed or rolled back.
- Record-oriented files.

  SFS files can be created with entry-sequenced, relative, or clustered organizations. Files can be accessed by using primary or secondary indexes.
- Flexible storage management.

The data component of a file and its indexes can reside on different volumes, and thus on different disks. This independence gives administrators greater flexibility in controlling availability and performance.

- Import/export capability.

  SFS files can be stored and retrieved from a file, disk, or tape device and can be transferred between SFSs.

The SFS is a nonhierarchical file system. SFS files are independent of the operating system file system. Administrators access SFS files through *sfsadmin* commands.

To organize record-oriented data, the SFS offers multiple file layouts: entry-sequenced (also known as sequential), relative, and B-tree clustering.

The records of entry-sequenced files are stored in the order in which they were inserted into the files; that is, each new record is appended to the end of the file. This type of file is useful for keeping time-sequenced event records, such as audit trails.

The records of relative files are stored in an array of fixed-length slots; a new record can be inserted into a specific slot, into the first free slot in the file, or at the end of the file. Relative files are useful for direct access by slot number.

The records of clustered files are organized in a tree structure. Clustered files are useful for records that are to be accessed by a field or combination of fields as specified by the creator of the file.

Each SFS file has a primary index, which defines the key to be used to physically order the records. A key is a sequence of one or more fields to be used as a basis for retrieving a record. Each SFS file can have zero or more secondary indexes. A secondary index specifies one or more alternative fields to be used to retrieve a record. A secondary index does not change the physical order of a file′s records. Instead, it creates an alternative sequence in which the records of a file can be accessed according to values of certain fields. If a file′s secondary index is based on fields whose values are most frequently accessed by an application, use of the index can minimize the time needed to access a record.

Each SFS file has a primary area that contains user data and the primary index. A file can also have zero or more secondary areas, one for each secondary index. A file′s areas can reside on the same or on different volumes, and thus on different disks or partitions. Note that while a volume can contain many areas, a single area cannot span more than one volume. The flexibility of file area storage allows administrators to easily control availability and performance. For instance, administrators may want to mirror only the volume on which the primary index resides, since secondary indexes can easily regenerated. This avoids the cost of mirroring the entire file.

To prevent unauthorized access, Encina SFS takes advantage of the authentication tools embedded in DCE. It provides access control list (ACL) protection for services including its administrative interfaces (for example, file creation, file deletion, and file listing). Through the use of the ACL feature, it is possible to protect SFS operations such as read, insert, and update operations, as well as grant exclusive file access to a client.

The SFS also supports multiple secondary indexes, as well as both transactional and nontransactional access. A range of locking strategies is available to enable trade-offs between full transactional serialization and higher degrees of concurrency. Encina SFS also provides NLS, which includes the ability to handle multibyte characters and collation sequences (for sorting in other languages).

6.2 Planning Considerations for Encina

You will require the use of Encina Base and Encina Server. They form an integral part of the CICS for AIX application server. The choice of whether or not to use Encina SFS, PPC Executive, and the PPC Gateway will depend on your configuration and needs. Understand your requirements and then select accordingly. We provide some guidance below.

6.2.1 CICS Queue and File Management

CICS queue and file management concerns the management of data associated with user files, auxiliary temporary storage queues, intrapartition transient data queues, and locally queued ATI requests. This data can be managed either with Encina SFS or DB2 Version 2 (see Chapter 7, "Interactions with the DB2 Family of Products" on page 85). The decision is yours. We recommend that you become familiar with both methods of managing CICS queues and files and select the method most appropriate for your situation.

If you decide to use Encina SFS to manage CICS queues and user files, there are three possible options:

- Local SFS

  With this option, an SFS that is being used by a CICS for AIX region is located on the same machine as the CICS for AIX region (see Figure 22). CICS administration services are used for configuring the SFS, CICS queues, and user files. CICS facilities can be used to configure and start the SFS. With this option the Encina FLT mechanism can be used, improving performance.

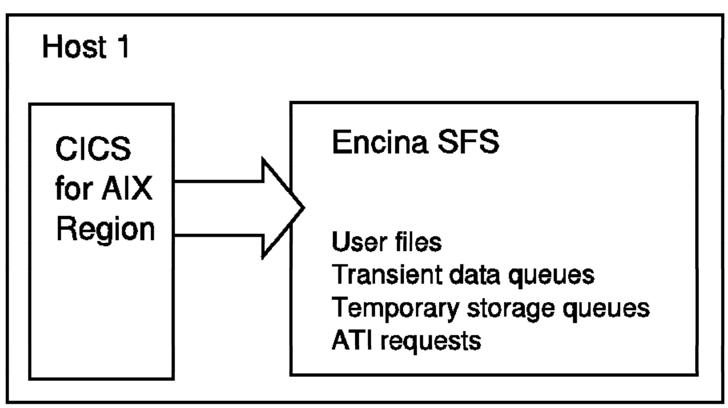

Figure 22. Using a Local SFS

- Remote SFS on a machine without CICS for AIX installed

  CICS for AIX regions use a remote SFS located on a machine that does not have CICS installed (see Figure 23 on page 82). Native Encina and DCE commands are required to configure the SFS, CICS queues, and user files. It is also necessary to use native AIX and Encina commands to configure and start the SFS.

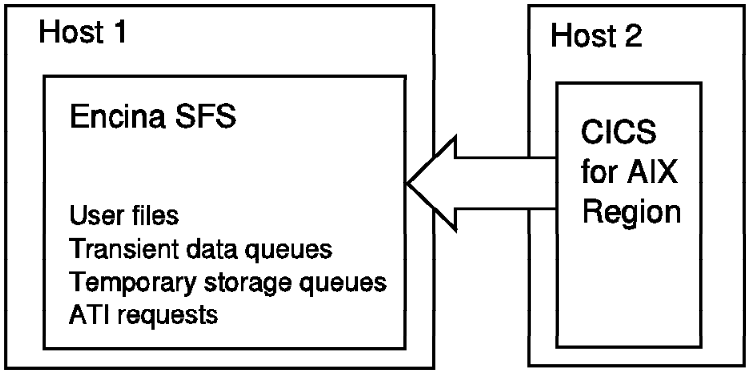

Figure 23. Using a Remote SFS without CICS for AIX Installed

- Remote SFS on a machine with CICS for AIX installed

  A CICS for AIX region on one machine shares an SFS with a CICS for AIX region on another machine (see Figure 24). CICS administration services are used for configuring the SFS, but native Encina and DCE commands are used to add the queues and user files to the SFS. With this option, native AIX and Encina commands rather than CICS facilities must be used to configure the SFS for CICS queue management.

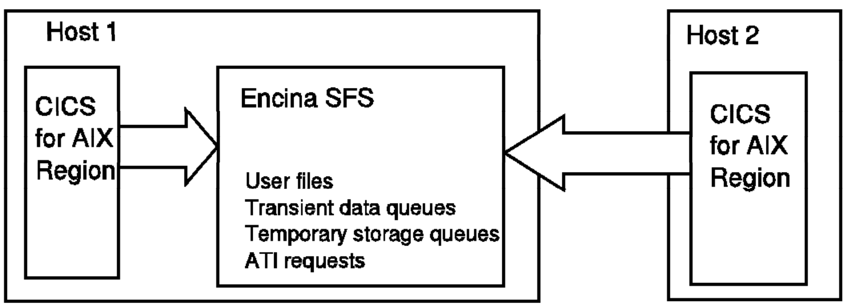

Figure 24. Using a Remote SFS with CICS for AIX Installed

6.2.2 DCE

Depending on your decision to use DCE, there are three possible cases to consider:

- Both the region and the SFS are in the same DCE cell

  In this case DCE can be used by CICS for AIX to locate the SFS, and it is possible to use user and RPC authentication.
- A CICS for AIX region that does not use the DCE CDS

  If the CICS for AIX region does not use the DCE CDS or security services, it is still possible for it to use an SFS that does use DCE CDS, as long as the SFS and the region are in the same TCP/IP network and the CICS for AIX region includes the SFS host name in the $CICS_HOSTS environment variable (held in /var/cics_regions/regionName/environment). However, both the region and the SFS must be defined with protection levels set to *none*. In this case, RPCs cannot be authenticated. See also 5.2.2, “Configuration without DCE CDS and Security Services” on page 73.

- Neither the CICS for AIX region nor the SFS is in a DCE cell

  If neither the CICS for AIX region nor the SFS are in a DCE cell, but they are both in the same TCP/IP network, they can communicate provided that the:
  - CICS for AIX region lists the SFS′s host name in the $CICS_HOSTS environment variable
  - Server binding information has been made available for clients.

  Both the CICS for AIX region and the SFS must be defined with protection levels set to *none*. In this case RPCs cannot be authenticated.

6.2.3 Intercommunication

Encina provides services to enable CICS for AIX communication over either TCP/IP or SNA. Thus CICS for AIX can communicate with other CICS for AIX or CICS on Open Systems systems as well IBM mainframe CICS systems.

There are alternative methods for establishing CICS for AIX communication over TCP/IP or SNA to the Encina PPC products. They are not equivalent, however, and the method you use should cover your specific requirements.

CICS for AIX supports two types of communication over TCP/IP. These are CICS family TCP/IP and Encina PPC TCP/IP. For a comparison of the two methods, see 3.3.2, “TCP/IP” on page 32.

CICS for AIX supports two types of communication over SNA. These are Local SNA support and SNA support using the Encina PPC Gateway server. For a comparison of the two methods, see 3.3.3, “SNA” on page 34.

The protocol and the implementation that you select will determine whether the Encina PPC components are required. See 3.4.6.1, “Selecting a Protocol” on page 42 for more considerations about which protocol to select.

6.3 Summary

Encina offers essential services for CICS for AIX through Encina Base and Encina Server. These services, which provide the base for the transactional services, are essential to the data integrity facilities of CICS for AIX

Encina SFS provides comprehensive file handling capabilities with a number of different file organizations. Encina SFS is most suited to record-based processing. It can be used to manage CICS for AIX queues and files. Use of Encina SFS is optional and will depend on your requirements.

Encina PPC Executive and Encina PPC Gateway provide communications-enabling software. Whether the PPC components are used in your configuration depends on your CICS for AIX configuration decisions. For example, if you need synchronization level 2 communication over TCP/IP, Encina PPC Executive will be required. If you require synchronization level 1, you may opt to use CICS family TCP/IP instead. The use of these PPC components enables communicating over TCP/IP or SNA to other CICS for AIX, CICS on Open Systems, or IBM mainframe CICS regions. The most significant benefit that the PPC components bring to CICS for AIX intercommunication is two-phase commit processing over an SNA link; indeed they are the only components that enable such processing.

Chapter 7. Interactions with the DB2 Family of Products

In this book CICS for AIX is the application server of choice. CICS for AIX provides the environment in which applications run. Those applications perform the necessary business logic; that is, the reading and writing of enterprise data. Data is accessed by applications at a logical rather than a physical level.

CICS for AIX provides access to four different types of data storage facilities: files services, queue services, relational database services, and journal services (see 3.1.4, “Access to Data” on page 27). In deciding which of these data storage facilities to use to store your application data, you must consider the following major issues: data structure, management functions, recovery of lost data, data sharing, distribution capabilities, integrity of shared data, and performance.

The use of an RDBMS is increasingly regarded as the most effective solution that addresses all of these issues. An RDBMS has knowledge of the logical and physical structure of the data. The structure information is held separately from the application. Thus the data structure can be changed significantly without any need to change the application. An RDBMS provides services and utilities for managing the recovery, sharing, and distribution of data. This presents a significant step forward for distributed processing as the inability to share data has historically been an impediment to application growth and change. Current hardware and software now make it practical to distribute a single database among several remote locations. Distribution allows you to keep data local to the code that most often uses it without restricting access to it by other users. An RDBMS maintains the integrity of the data under its control.

The integrity of data must be maintained as it is accessed and updated from a CICS application. Ensuring data integrity requires the use of an interface between CICS for AIX and any RDBMS in which data is accessed. The interface must be capable of coping with decisions as to whether to commit updates or back them out. Such an interface must also be able to cope with the fact that data has been updated in a location other than the CICS for AIX region or RDBMS. Ideally the interface should also conform to a standard so that any RDBMS conforming to the standard can be connected to CICS for AIX. Having several RDBMSs from which to select gives the enterprise more flexibility in combining CICS for AIX with the RDBMS of its choice. In order to achieve this level of data integrity and flexibility, database access is provided through the X/Open XA interface. The use of the X/Open XA interface requires support in both CICS for AIX and any RDBMS to which it connects. The X/Open XA interface is covered in more detail in 7.1, “X/Open XA Interface” on page 86.

At the time of writing the RDBMSs supported by CICS for AIX Version 2.1.1 are DB2 for AIX V1.1 with appropriate program temporary fixes (PTFs), DB2 for AIX V2.1, Informix V5.0.5, Oracle V7.1.6, and Oracle V7.2.2. We focus on the DB2 family of products, which is a collective name for IBM RDBMS products available for the mainframe, Intel-, and UNIX-based platforms.

The DB2 family of products has a very high level of interconnectivity in the same way that CICS does. The versions of DB2 that run on AIX, OS/2, and Windows NT belong to the DB2 common server. DB2 common server is the generic name for DB2 products running on Intel- or UNIX-based platforms. They provide a common set of functions and connectivity among themselves.

In this chapter we show how it is possible to access and update data held in each of the DB2 family members from a CICS for AIX application, thereby demonstrating the great flexibility that is available to the application developer. Data held on a variety of platforms can be made accessible to a CICS for AIX application and not just data held local to the application.

7.1 X/Open XA Interface

The X/Open XA interface is a standard, industrywide specification of how updates that involve an application program, a transaction manager, and one or more resource managers, can be coordinated with full integrity. Software products conforming to the X/Open XA interface can be combined to provide full update integrity for any data updated by the application program. Thus a CICS for AIX application accessing data in a DB2 for AIX database through the X/Open XA interface will have full update integrity for any updates that take place. In the CICS for AIX implementation of the X/Open Distributed Transaction Processing (X/Open DTP model) standard, the transaction manager (CICS) coordinates transaction initiation and completion among the resource managers (RDBMSs) as initiated by a CICS application program.

It is important to understand the X/Open DTP model before we discuss the CICS for AIX implementation of it.

7.1.1 X/Open DTP Model

Figure 25 shows the basic X/Open DTP model for a transaction processing environment.

The boxes represent the functional components; the connecting lines represent the interfaces between components. The arrows indicate the directions in which control may flow. Data may flow in both directions across each interface.

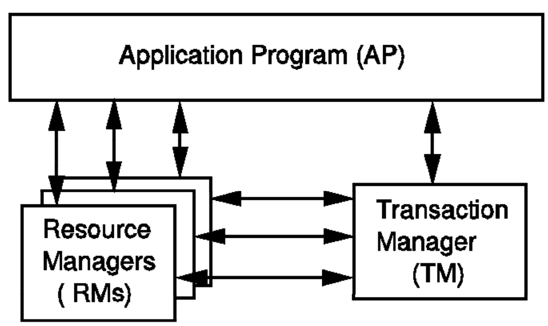

Figure 25. X/Open DTP Model: Functional Components and Interfaces

For a description of each functional component, see 7.1.1.1, “Functional Components” on page 87. For a description of the three interfaces shown, see 7.1.1.2, “Interfaces among Functional Components” on page 87.

The functional components are not necessarily separate processes, nor are they necessarily totally within a single process.

7.1.1.1 Functional Components

The three functional components of the X/Open DTP model are the application program, transaction manager, and resource manager.

Application Program: The application program (AP) implements the desired function for the enterprise. Each AP specifies a sequence of operations that involves resources such as databases. An AP defines the start and end of a global transaction, accesses resources within transaction boundaries, and usually decides whether to commit or roll back each transaction.

Transaction Manager: The transaction manager (TM) manages global transactions and coordinates the decision to commit them or roll them back, thus ensuring atomic transaction completion. The TM also coordinates recovery activities of the resource managers when necessary, such as after a component fails.

Resource Manager: The resource manager (RM) manages a certain part of the computer′s shared resources. Many other software entities can request access to the resource from time to time, using services that the RM provides. Examples of RMs include a database management system (DBMS), a file access method, such as X/Open indexed sequential access method (ISAM), or a print server. Some RMs manage a communication resource.

In the X/Open DTP model, RMs structure any changes to the resources they manage as recoverable and atomic pieces of work. The RMs let the TM coordinate completion of the transactions atomically with work done by other RMs.

7.1.1.2 Interfaces among Functional Components

There are three interfaces among the functional components in the basic X/Open DTP model:

AP-RM: The AP-RM interfaces give the AP access to the shared resources managed by the RM. Existing X/Open interfaces, such as SQL and ISAM, provide AP portability. The X/Open DTP model imposes few constraints on the native RM API. X/Open may specify additional, specialized AP-RM communication interfaces for DTP, such as peer-to-peer and RPC.

AP-TM: The AP-TM interface (the TX interface) lets the AP delimit global transactions. The TM provides routines that let the AP start and complete global transactions. The TM completes global transactions on the basis of a request from the AP and coordinates with the participating RMs and other TMs. When this coordination is completed, the TM returns the completion status of the AP. Details of the AP-TM interface are in *X/Open Distributed Transaction Processing: The TX (Transaction Demarcation) Specification*.

TM-RM: The TM-RM interface (the XA interface) lets the TM structure the work of RMs into global transactions and coordinate global transaction completion and recovery. In the XA specification, the routines that each RM provides for the TM′s use are the **xa**_routines. The routines the TM provides for the RMs to call from are the **ax**_set routines.***:*** When an AP calls a TM through the TX interface, the TM typically implements each TX call by contacting RMs through the XA interface. Because the XA interface is invisible to the AP, the TM and RM can use other methods to interconnect without affecting application portability.

A TM assigns a data structure called a *transaction identifier* (XID). The XID lets the TM track and coordinate all of the work associated with a global transaction. Each RM maps the XID to the recoverable work it did for the transaction. For global uniqueness, the XID should contain atomic action identifiers.

Activity among Functional Components: Activity among the three functional components of the X/Open DTP model is dependent on the transaction operation.

Transaction Initiation: When an AP instructs its TM to start a global transaction, the TM tells all appropriate RMs to associate the information about the global transaction with any work the AP may request from them.

Some RMs are configured so that the TM does not inform them when a global transaction starts. The RM contacts the TM to become associated with a global transaction only after the AP calls it to request actual work. This is called *dynamic registration*. If *static registration* is requested, each new transaction in the process generates a call to the RM that tells it that it has joined a new transaction or resumed an old one.

Transaction Commitment: When an AP instructs its TM to commit a transaction, the TM and RMs use two-phase-commit presumed rollback to ensure that the transaction′s updates are atomic.

In phase 1, the TM asks all RMs to *prepare to commit* (or *prepare*) their work. It also asks whether the RM can guarantee its ability to commit the work it did on behalf of a global transaction. If an RM can commit its work, it replies affirmatively. A negative reply reports failure.

In phase 2, the TM directs all RMs either to commit or to roll back the work done on behalf of a global transaction, as the case may be. All RMs commit or roll back changes to shared resources and then return status to the TM.

When an AP calls its TM to commit a global transaction, the TM reports on whether commitment or rollback was the outcome. This report is based on reports the TM received (directly or through other TMs) from all involved RMs.

The XA specification contains two optimizations in the calling sequence between the TM and RM. An RM can withdraw from further participation in a global transaction during phase 1 if it was not asked to update shared resources (the *read-only optimization*). A TM can use one-phase commit if it is dealing with only one RM that is making changes to shared resources.

The XA specification discusses requirements for stable recording of transaction data, including specifying when the TM and RMs are free to discard their knowledge of the global transaction.

Transaction Rollback: The TM rolls back the global transaction if any RM responds negatively to the phase 1 request or the AP directs the TM to roll back the global transaction.

The TM effects phase 2 by telling RMs to roll back the transaction. The RMs must not let any changes to shared resources become permanent.

Heuristic Transaction Completion: In certain, unusual, cases, the RM could experience a long delay between phases 1 and 2 of the two-phase-commit protocol. For example, the TM that issued the prepare-to-commit (phase 1) request could block or fail later in the protocol sequence. To free resources, an RM can complete its work *heuristically* (independent of direction from its TM). The heuristic decision can be prompted by administrative action or by completion of a parametric timeout interval.

Heuristic decisions typically cannot depend on the knowledge of results at other RMs that TMs normally coordinate. When any RM makes a heuristic decision, the global transaction may fail to maintain global atomicity; one RM may commit its work while another rolls back its work. Such *mixed-heuristic* completion can leave shared resources in an inconsistent state. A TM can report a mixed-heuristic condition to its AP.

Failures and Recovery: Recovery is a process of restoring shared resources to a consistent state after various types of failure. The X/Open DTP model makes these assumptions:

- TMs and RMs have access to stable storage.
- TMs initiate and control transaction recovery.
- RMs provide for their own restart and recovery as directed by TMs.

7.1.2 CICS for AIX and the X/Open DTP Model

As a result of the implementation of the X/Open XA interface within CICS for AIX the system administrator has to provide information about which databases are to be accessed. In this section we explain how that information is supplied and show how the XA interface is activated.

7.1.2.1 Architecture

Figure 26 shows the architecture of CICS for AIX Version 2.1.1 in relation to the X/Open DTP model.

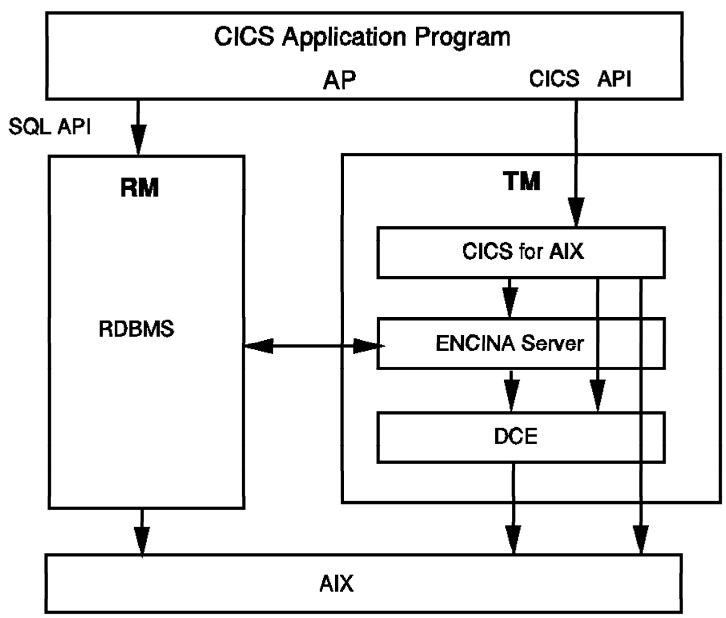

Figure 26. CICS for AIX and the X/Open DTP Model

From Figure 26 you can see that an application program communicates with the transaction manager component through the EXEC CICS commands. The program can also communicate with the resource manager, an RDBMS, using the EXEC SQL commands.

Communication between the transaction manager component and the resource manager is achieved through services provided by Encina Server. This does not affect the application program, which continues to interact with CICS for AIX only. The TX interface is hidden from the application program.

7.1.2.2 Definition

In this section we show the components that are involved in defining an activating X/Open XA interface to an RDBMS from the CICS for AIX perspective. There are several aspects to defining and enabling the interface:

- XA resource definitions
- CICS for AIX region environment setup
- CICS for AIX region startup

XA Resource Definitions: CICS for AIX uses product definitions to hold the information that it needs to interface to other transactional products. Each XA definition (XAD) entry contains information for one product. These entries are located in the *XAD.stanza* file in the */var/cics_regions/<region_name>/database/XAD* directory, where *<region_name>* is the name of the CICS for AIX region.

Each XAD entry consists of several attributes. The attributes relevant to our discussion are:

<Key> Represents the name of the product and is the key for the XAD entry. Product names can be up to 12 characters long.

SwitchLoadFile Switch load file path name. This ASCII text attribute is a path name to an object file (linked using ld but not a main() program) that contains the xa support subroutines for this XA-compliant product. The default value is ″″. By using this switch load file, CICS for AIX can invoke the appropriate subroutines of the RDBMS, for example, DB2 for AIX.

XAOpen Resource manager initialization string, an ASCII character string that is passed to the XA-compliant product′s xa_open() function. It is specific to the XA product being defined. The default value is ″″.

XAClose Resource manager termination string. This is an ASCII character string that is passed to the XA-compliant product′s xa_close() function. The layout of the string is specific to the XA product being defined. The default value is ″″.

XASerialize Resource manager serialization attribute. Indicates how CICS for AIX should serialize access to an XA-compliant RM in a multithread process. This attribute is used to indicate that an XA-compliant resource manager supports xa calls from multiple threads and which style of serialization the XA-compliant product requires. The default value is *all_operations*. The following values are accepted:

- **all_operations** CICS for AIX serializes around each xa call made. This is the default setting.
- **start_end** CICS for AIX serializes around xa_start and xa_end calls. Serialization takes place around a transaction.
- **single_association** or **multiple_association** Allows CICS for AIX to serialize on the basis of whether the XA-compliant product is thread aware or not.

CICS for AIX Region Environment Setup: The *environment* file in the */var/cics_regions/<region_name>* directory, where *<region_name>* is the name of the CICS for AIX region, contains a list of environment variables that are set before the startup of the region. These variables may be specific to the resource manager to which you are connecting. For example, the **DB2INSTANCE** variable indicates the name of the instance of DB2 for AIX to which the CICS for AIX region will connect.

CICS for AIX Region Startup: In the first phase of the startup the main *cics* process initializes several other cics processes including the application manager (*cicsam*). Subsequently the CICS application servers (*cicsas*) are created by cicsam. As part of the XA interface registration, the XA definitions are scanned to obtain the switch load file name and the XA open string of the resource managers. The switch load file is an object file. It is loaded into each cicsas through the AIX *load()* system call. Each cicsas process then initiates an *xa_open()* call using the XA open string specified in the region′s XAD stanza file followed by an *xa_recover()* call, which causes the resource manager stub to identify all xa routines and connect to the database. The resource manager then completes the registration process. The work performed depends on the resource manager and its configuration. For example, in the case of DB2 for AIX, a connection is established with the DB2 for AIX server for local configuration. For a remote configuration, the machine name is located, and the corresponding port number and remote host name are found. A connection is then established with the DB2 for AIX server on the remote host name system.

7.2 RDBMS Access Options

In this section we examine a number of configuration issues to consider when implementing a connection from CICS for AIX to an RDBMS. The issues we cover may influence your choice of an RDBMS. The first issue is to fully understand what happens when a database is accessed in an XA or non-XA mode. We then discuss a possible performance optimization that could help reduce the overhead of using the XA interface.

7.2.1 XA-Enabled and Non-XA-Enabled Databases

With an RDBMS that supports the XA interface, CICS and the resource manager can exchange transactional information and hence enable coordinated commitment or rollback of the resource changes. The resources involved may be in both CICS and in an RDBMS database. The resources are always maintained in a consistent state. Such RDBMS databases are called *XA-enabled relational databases*. DB2 for AIX is a good example of an RDBMS that supports the XA interface. An XAD resource definition is required for a CICS for AIX region that accesses an XA-enabled relational database (see 7.1.2.2, “Definition” on page 90).

With an RDBMS that does not support the XA interface, there is no transaction coordination between CICS and the resource manager. If a transaction that makes changes to various CICS resources (such as CICS queues or files), in addition to changes to tables in an RDBMS, fails, data may be inconsistent. Intervention is required to check the consistency of data and to resynchronize the various resources. Such RDBMS databases are called *non-XA-enabled relational databases*. You do not have to define an XAD resource definition in this case. The absence of an XAD entry for the database indicates that the database will be accessed as a non-XA database even though the RDBMS may be XA capable.

Figure 27 shows the relationship between CICS and a resource manager for both an XA- and a non-XA-enabled database. In the case of the XA-enabled database, transactional information is exchanged between CICS and the resource manager. There is no such exchange for the non-XA-enabled database.

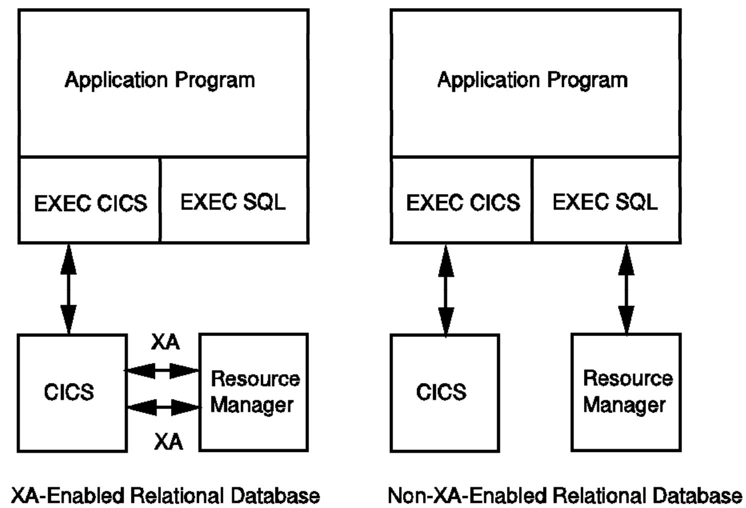

Figure 27. XA-Enabled and Non-XA-Enabled Relational Databases

If an XA-enabled database connection has been defined in a CICS for AIX region, that connection will always be used. It is not possible to deactivate the use of the XA interface once the region has started. If you want to use a non-XA definition, the entry must be removed from the XAD stanza file, and the CICS for AIX region must be restarted.

Simultaneous access to an XA-enabled database and a non-XA-enabled database from a CICS for AIX region is not supported, that is, only databases defined in the XAD entry for a CICS region can be accessed by a transaction running within that region.

You can use a non-XA connection to the second database concurrently with an XA connection to the first database, that is, one-phase and two-phase supporting databases can be accessed concurrently within a transaction, but the one-phase supporting database is restricted to read-only access. CICS establishes implicit connection to the first database through the XA interface according to the XAD entry during CICS startup time. To connect to the second database, you have to connect explicitly by using an EXEC SQL CONNECT call. To reconnect to the first database, you have to connect to it explicitly by using an EXEC SQL CONNECT call even though the first database is defined in the XAD entry.

Table 5 on page 93 compares an XA-enabled relational database and a non-XA-enabled relational database. The resource manager is DB2 for AIX.

Table 5. XA-enabled and Non-XA-Enabled Relational Databases

XA-Enabled Relational Database	**Non-XA-Enabled Relational Database**
Transactional information exchange between CICS for AIX and DB2 for AIX takes place.	Transactional information exchange between CICS for AIX and DB2 for AIX does not take place.
Coordinated commitment or rollback of both CICS resources and RDBMS table changes occurs through the EXEC CICS SYNCPOINT call.	CICS application has to issue EXEC SQL COMMIT call or EXEC SQL ROLLBACK call explicitly for the changes of RDBMS tables. Only the CICS resource changes are affected by the EXEC CICS SYNCPOINT call.
An XAD entry (Product Definition) is required for each database. CICS for AIX has to be cold started after any changes to the XAD stanza file or the addition of new XAD stanza file entries.	An XAD entry is not required. A CICS application that wants to connect to a particular database has to issue an EXEC SQL CONNECT call.
Provides better performance because CICS transactions can reuse an XA-managed database connection while the application server (cicsas) is running, without having to call EXEC SQL CONNECT repeatedly.	CICS applications have to issue the EXEC SQL CONNECT call to start the database connection for each transaction. Hence there is a greater overhead.
CICS immediately abends the region if it cannot open a connection to the database.	CICS region keeps running when the EXEC SQL CONNECT call issued by a CICS application fails.
CICS facilities for transactional control must be used. These are the EXEC CICS SYNCPOINT and EXEC CICS SYNCPOINT ROLLBACK commands.	RDBMS facilities must be used. They are the EXEC SQL COMMIT, EXEC SQL ROLLBACK, and EXEC SQL SYNCPOINT commands.
CICS SYNCPOINT processing takes longer than an SQL COMMIT call because it includes coordination of syncpoint among all resources managers involved.	SQL COMMIT processing is faster, but it only works if no other resource managers are involved in the transaction. Furthermore, using SQL COMMIT in a transaction prevents you from issuing an EXEC CICS SYNCPOINT call in the same transaction.
CICS application programs must be compiled with the NOSQLINIT option.	Special compile options are not required.
Database connection stays with the application server (cicsas). Hence it is conveyed across an EXEC CICS LINK or EXEC CICS XCTL call.	Database connection cannot be conveyed across the EXEC CICS LINK or EXEC CICS XCTL call. You must disconnect from the database before these calls and reconnect in the called program.
Database connection is established according to the XAD definitions at CICS startup time. There is no need for the CICS transaction to make the database connection.	CICS transaction has to connect to the database through an EXEC SQL CONNECT call before making any SQL access calls.

7.2.2 Single-Phase and Two-Phase commit

Some CICS for AIX configurations do not require the full data integrity provided by the two-phase commit protocol of the XA interface. Where multiple resource managers are involved in a transaction but only one of them has updated a resource, those that have not updated a resource do not have to go through syncpoint processing. (After all, they have no updates to commit or roll back.) This optimization is called *single-phase commit*, and it is available in DB2 for AIX.

Single-phase commit can improve performance, but the limitation of one database may restrict its use. No resources outside DB2 for AIX should be updated. Therefore recoverable CICS queues and files or any CICS intersystem communication at synclevel 2 cannot be updated.

Table 6 contains a comparison between single-phase and two-phase commit.

Table 6. Single-Phase and Two-Phase Commit

Single-Phase Commit	**Two-Phase Commit**
Consistent data when DB2 for AIX is the only resource manager participating in the CICS transaction. Otherwise data integrity cannot be guaranteed.	Full data integrity is maintained for all resource managers involved in the CICS transaction.
The database under the single-phase commit optimization does not participate in CICS recovery. It may be necessary to manually re-synchronize the database with other resource managers.	The database fully participates in the CICS recovery process.
A CICS for AIX region can support at most one database under the single-phase commit optimization. Concurrent support of an XA-enabled DB2 database is not allowed.	A CICS for AIX region can support multiple XA-enabled databases concurrently.
Shorter syncpoint processing time because only one resource manager, DB2 for AIX, is invoked to commit or roll back changes to data. The prepare phase is skipped, hence the name *single-phase commit*.	Longer syncpoint processing time because all resource managers participating in the CICS transaction are invoked to perform commit or rollback processing.
XAD entry is required. The switch load file is *cics1pcdb2* in the /usr/lpp/cics/bin directory.	XAD entry is required. The switch load file is *cicsxadb2* in the /usr/lpp/cics/bin directory.

7.2.3 Dynamic and Static Registration

Registration is the process of a transaction manager informing a resource manager about a new global transaction. RDBMSs differ in the type of support that they offer. Registrations apply only to XA-enabled relational databases. There are performance and recovery implications when you select an RDBMS that supports a particular type of registration.

A dynamically registering XA-enabled relational database such as DB2 for AIX receives XA calls only when SQL commands are issued. Thus for those transactions that do not issue SQL commands, no XA calls are made. If the relational database suffers a catastrophic failure, CICS transactions running against the XA-enabled database will fail. CICS only catches the database failure if a new CICS application server is started. CICS must be shut down before the database is brought back online after recovery. If you attempt an automatic start of the CICS region before the database has been recovered, the region abends.

With statically registering XA-enabled relational databases such as Oracle and Informix, all CICS transactions in the region will make XA calls even though some transactions do not have to access the database. When the relational database suffers a catastrophic failure, CICS catches the failure and immediately abends the region. You cannot warm start the CICS region until the database has been recovered, otherwise the region will abend.

If the majority of your CICS transactions do not have to access a database, we strongly recommend that you use a database that supports XA dynamic registration such as DB2 for AIX. In this way, CICS can arrange to drive syncpoint in the database only when the database is actually updated for a given transaction. This can reduce the time it takes to syncpoint a transaction when running in an environment with multiple XA databases and CICS control files, where transactions usually only update data managed by some of the applications.

Table 7 compares dynamic and static registration.

Table 7. Dynamic and Static Registration

Dynamically Registering Relational Database	**Statically Registering Relational Database**
Only a CICS application that issues SQL calls makes an XA call.	Every CICS application makes an XA call.
In the case of catastrophic failure of the database, CICS transactions accessing that database fail. The other CICS applications continue to run. CICS abend is deferred until the next application server (cicsas) is started and CICS detects the database failure during opening of the connection.	In the case of catastrophic failure of the database, the CICS for AIX region immediately abends. The database has to be recovered before CICS can be started up again.
Shorter syncpoint processing time because the resource manager is invoked only to commit or roll back when the data it manages changes	Syncpoint processing time is longer because the resource manager is invoked every time regardless of whether the data it manages changes or not.

7.3 Using DB2 for AIX as CICS File Manager

With CICS for AIX Version 2.1 and later, you can use a DB2 for AIX database for CICS file management instead of a local or remote Encina SFS. CICS file management refers to the management of data associated with user files, auxiliary temporary storage queues, intrapartition transient data queues, and locally queued ATI requests. In simple terms, this enhancement is a mapping function of the CICS for AIX SFS files to DB2 for AIX tables. CICS applications maintain the same logical views of these files, that is, as though they were SFS files. In reality the data is stored in the underlying DB2 tables that belong to the default XA-enabled DB2 for AIX database for the CICS region.

CICS for AIX provides a complete set of administrative utilities to:

- Migrate an existing CICS region to use DB2 as a file manager
- Configure a new CICS region to use DB2 as a file manager
- Create CICS tables on DB2 from SFS schema files containing a description of the CICS file characteristics
- Create and populate CICS tables on DB2 from a flat file format
- Provide access to these mapped files from COBOL batch programs

In a CICS for AIX configuration where DB2 for AIX is used as a resource manager and relational databases are accessed only through the XA interface, administrative costs can be reduced because you no longer have to manage the Encina-SFS-related items. If the following conditions are true:

- The application program does not make updates through the Encina PPC Gateway
- DB2 for AIX is the only database resource manager
- All of your application data and CICS data are on the same database

syncpoint processing performance can be improved through the use of the XA single-phase commit optimization or non-XA native SQL commit processing. Results from performance measurements for CICS for AIX indicate that CICS mapped files on DB2 using the single-phase commit optimization perform better than the best case of CICS files on SFS.

Be aware of the following limitations and restrictions when you use a DB2 database server as the CICS file manager rather than SFS:

- The CICS region uses a single database for all user files and CICS queues. This is the database specified in the RD DefaultFileServer attribute.
- Because of the limitation imposed by holding files in a relational database, some facilities of the COBOL external file handler are not supported. These include:
  - Mixing transactional access and nontransactional access to files in a single application is not permitted.
  - When there is nontransactional access to a file, only single record locking is enabled.
  - The facility to dynamically add indexes to a file at run time is not supported.
- Nonrecoverable CICS queues are supported as recoverable facilities on DB2. This affects the record locking characteristics of these facilities. The change in semantics means that data is not written to a file until a syncpoint is taken. If your CICS applications are dependent on nonrecoverable file semantics, we recommend that you use an Encina SFS for CICS queue and file management. The following CICS resources are maintained as DB2 recoverable resources:
  - Nonrecoverable user files
  - Nonrecoverable auxiliary temporary storage queues
  - Locally queued unprotected START requests
  - Physically recoverable transient data queues
  - Nonrecoverable transient data queues

- SFS fields migrated to DB2 for AIX are not mapped in the DB2 schema definitions. SQL online query tools only show raw data and cannot map the fields.
- There is no access control to the mapped files similar to RPC authentication and DCE access control lists (ACLs).

  The granularity of access to RDBMS from CICS is at the region level, with all online applications running in CICS application server (cicsas) processes accessing a DB2 for AIX database with the same database userid that is authenticated at cicsas (xa_open) initialization time.

  You can control access to these mapped files by implementing CICS transaction security and CICS resource security in the CICS for AIX region.

Where DB2 for AIX Version 2 is integrated with a CICS for AIX region, CICS applications can use the SQL CONNECT command to connect to multiple DB2 databases within the same unit of work. The DB2 database to which CICS files are mapped as the default database requires an SQL CONNECT RESET command to reset the connection to this default database before any dynamic SQL calls can be issued to serve a CICS file control, temporary storage, or transient data request.

Figure 28 on page 98 shows how user CICS applications should be coded to switch the access from one DB2 database to another one and reconnect back to the default database before a CICS syncpoint is taken to commit all database changes.

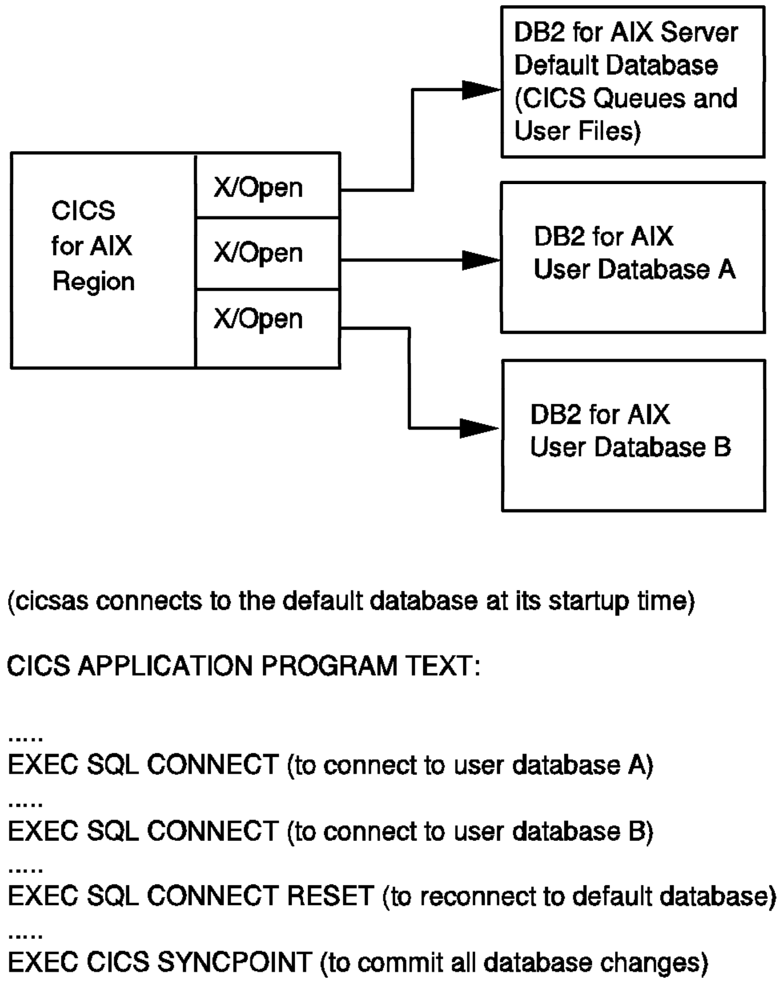

Figure 28. Switching Access among DB2 for AIX Databases

7.4 Planning Considerations for DB2 Family Access

In this section we explain how each DB2 family product can be accessed by a CICS for AIX application and discuss any planning considerations relevant to each product.

7.4.1 Accessing a DB2 for AIX Database

In this section we look at how DB2 for AIX databases can be accessed by a CICS application running in a CICS for AIX region. We restrict our discussion to XA-enabled databases only. There are two possible configurations for CICS for AIX to DB2 for AIX communication: local and remote.

7.4.1.1 Local Configuration

This is the simplest configuration for an application in a CICS for AIX region to access DB2 databases. Both the CICS for AIX region and the DB2 for AIX server reside in the same RISC System/6000 machine. IBM DB2 Server for AIX must be installed on the local machine, and an instance of a DB2 for AIX server has to be created on that machine.

At CICS startup time, each CICS application server process (cicsas) establishes the XA connection using the XA open string obtained from the XAD stanza file entry. See *CICS on Open Systems Administration Reference*, SC33-1563-01 for more details. It is important to define the XAD stanza file entry correctly, particularly the path location of the switch load file and the XA open string. If the cicsas cannot connect to the database because either the database is not active or because the XAD stanza file entries are incorrect, the CICS for AIX region will abend at startup.

Figure 29 shows the local implementation of a DB2 for AIX database being accessed from a CICS for AIX region.

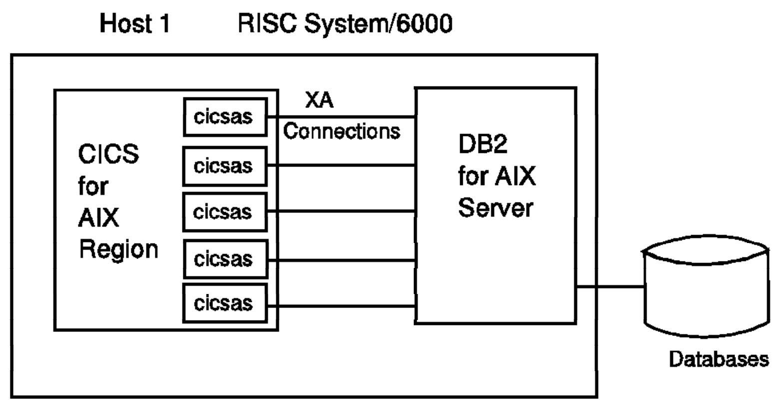

Figure 29. CICS Access to Local DB2 for AIX Database

As DB2 for AIX uses a database name in the XA open string, you have to create the database before region startup.

Switch Load File: CICS for AIX provides two versions of the switch load file: cics1pcdb2 for single-phase commit and cicszadb2 for two-phase commit. Select the version that is appropriate for your configuration.

Regardless of whether the DB2 for AIX databases are accessed using a local or remote configuration, the entries in the XAD stanza file for the databases will be the same; that is, there are no location-specific parameter considerations.

We do not recommend that you create your own version of the switch load file.

CICS for AIX Region Environment Setup: In the */var/cics_regions/<region_name>* region directory, the *environment* file is used to set environment variables on region startup.

The environment variable for DB2 for AIX is DB2INSTANCE. This variable specifies the instance name of the DB2 server to which CICS is to refer.

As there is only one DB2INSTANCE variable, a CICS for AIX region can only connect to the databases of that instance.

Shared Object: You have to create a DB2 for AIX shared object code. As its name suggests, the shared code is loaded into memory once in the shared library segment and shared by all processes that reference it. The CICS for AIX COBOL run-time environment, CICS for AIX C transactions, and the switch load file must reference the DB2 for AIX shared object at run time. See *DB2 for AIX*

Planning Guide, S20H-4758, and *DB2 for AIX Installation and Operation Guide*, S20H-4757, for more details.

Number of XAD Entries: You can create multiple databases running under one instance of the DB2 for AIX database manager. For each database, CICS for AIX requires an XAD entry with the name of the database as the connection string (XA open string). So, there should be as many XAD entries as there are databases being accessed by the application.

A CICS application can switch from one database to another explicitly, using the SQL CONNECT command. If you do not explicitly connect to a database in your application, CICS for AIX assumes that the database the application wants to access is the one that was last successfully connected to.

It may be possible to physically merge multiple databases into a single database, in which case the run-time overhead of database switching in your CICS applications can be saved. Whether this is practical or not will require detailed knowledge about the usage profiles of the databases used by your CICS applications. The backup and recovery requirements within DB2 for AIX or business requirements may be overriding obstacles against such a merge, however.

Number of Database Accesses: There is typically a noticeable overhead for crossing between a CICS for AIX application program and a relational database, so the number of such crossings should be minimized for performance reasons. The use of static SQL or SQL stored procedures can assist in reducing the overhead due to crossing between the CICS application program and the relational database.

Security Implications: It is important to understand the security implications when accessing relational database resources from CICS in an XA-enabled environment. The granularity of relational database access from CICS in this type of environment is at the region level. All transactions that run against an XA-enabled database access that database with the same database userid, that of the CICS for AIX region.

This is an implication of the XA architecture under which resource manager authentication occurs. For CICS, the xa_open call is issued during application server initialization. At this point, a connection is established to the relational database with a specific database userid.

Each transaction that runs on this particular application server inherits the established database connection and, by implication, the established database authorization. Each CICS application server in a particular region connects to the database with the same database userid (because it uses the same xa_open() string). Therefore, each and every transaction that makes EXEC SQL calls makes them under the authority of this specific database userid. There is no mechanism for CICS to provide DB2 for AIX with the actual userid that invokes the CICS transactions.

A security policy in an XA-enabled CICS environment should therefore be based on CICS security facilities. CICS has facilities to control user access to the system as well as to specific resources, such as the transactions. You should consider implementing CICS transaction security and CICS resource security to limit access to those transactions.

DB2 has the concept of binding a program (or package) to a database. For CICS transactions that use static SQL statements, the binder′s privileges are used to access the database. Because all transactions execute under the same database userid, this userid must still be granted EXECUTE privileges against packages that involve CICS transactions. CICS transactions that use dynamic SQL have their access authentication done at run time. For more information about DB2 security in an XA environment, see the *DATABASE 2 for AIX Administration Guide*, S20H-4580.

The local configuration is probably the easiest to implement, but the worst from a resource distribution point of view. For best throughput the database machine should be located on a dedicated machine. With a local configuration, the database server and CICS for AIX region compete for system resources such as processor cycles and memory.

7.4.1.2 Remote Configuration

In a remote configuration the CICS for AIX region and the DB2 for AIX server reside in two separate RISC System/6000 machines. IBM DB2 Server for AIX must be installed on the remote database machine, and an instance of a DB2 for AIX server must be created there as well. In the local machine where the CICS for AIX region resides, DB2 Client for AIX must be configured. The Client Application Enabler for AIX (CAE) product provides the DB2 client code. CAE is a run-time environment for applications accessing a remote database server.

To connect a DB2 Client for AIX to a DB2 Server for AIX you use the remote data services (RDS) protocol, also known as DB2 common server private protocol. RDS supports distributed unit of work (DUW) with two-phase commit protocol.

A shared object, an XAD entry, and the switch load file are required on the local (CICS for AIX region) machine. The DB2INSTANCE environment variable must be set to the DB2 instance name and be present on both the local and the remote machines.

When the CICS application server (cicsas) issues the xa_open call to the local DB2 client, the database alias is obtained from the XA open string, and the DB2INSTANCE value is obtained from the CICS for AIX region environment file. Then the system database directory, for that DB2INSTANCE checked for a match in the database alias name. From the matched entry in the system database directory, the machine name is obtained and is checked against the machine names in the node directory. Once a match is found, the port number and the host name are obtained and checked against the /etc/services file of the host name for a port number match. If the port numbers match, a connection is established to the DB2 for AIX server on the host name system, which is the remote machine.

Figure 30 on page 102 shows a CICS for AIX region accessing databases in a remote DB2 system.

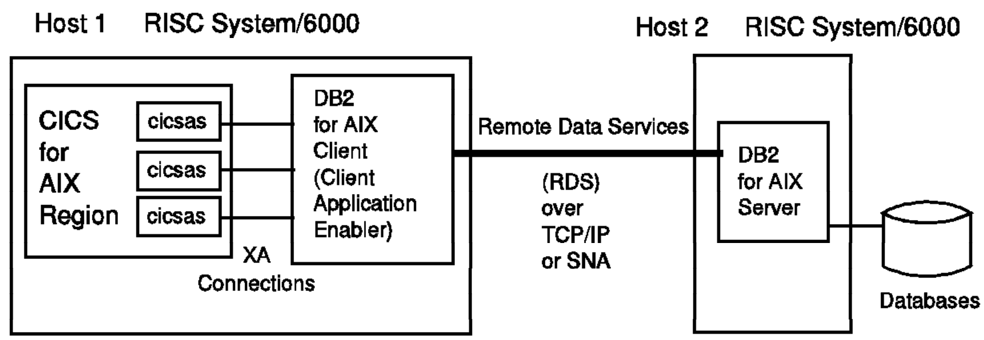

Figure 30. CICS Access to Remote DB2 for AIX Databases

The remote configuration is the most commonly chosen method for production because it provides a dedicated database machine. The DB2 for AIX server is not competing with the CICS for AIX region for system resources such as processor cycles and memory. This can lead to improved performance of the CICS for AIX region when compared with the local configuration (depending on how constrained the local configuration is).

The planning items for for the local configuration are also applicable to a remote configuration.

7.4.2 Accessing a DB2 for OS/2 or DB2 for Windows NT Database

Figure 31 shows a configuration in which a CICS for AIX region accesses one of the Intel-based versions of DB2, DB2 for OS/2, or DB2 for Windows NT. The CICS for AIX region resides in the local RISC System/6000 machine where the DB2 Client for AIX was configured after CAE was installed. DB2 Server for OS/2 or DB2 Server for Windows NT must be installed on the remote database Intel-based machine, and an instance of the DB2 server must be created there.

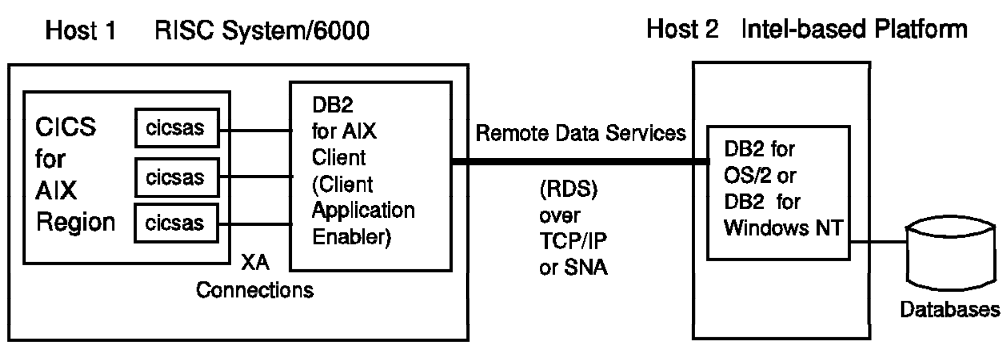

Figure 31. CICS for AIX Access to Remote DB2 for OS/2 or DB2 for Windows NT Database

A shared object, an XAD entry, and the switch load file are required on the local (CICS for AIX region) machine. The DB2INSTANCE environment variable must be set to the DB2 instance name, and it must be present on both the local and the remote machines.

7.4.3 Accessing a DB2 for MVS/ESA Database through DDCS

CICS for AIX can access enterprise data stored in DB2 for MVS/ESA databases through the use of DDCS for AIX. The DB2 for MVS/ESA database server is the DRDA application server; it is the only means of connecting the DB2 for AIX client to the DB2 for MVS/ESA server. The network protocol over which the connection occurs is restricted to SNA LU 6.2, and DDCS must be used. DDCS can be installed in a separate machine that sits between the local RISC System/6000 machine containing CICS for AIX and the DB2 for AIX client. For simplicity, we illustrate DDCS installed in the local machine.

DRDA is the foundation of IBM′s distributed database solution. It is a set of protocols that enable transparent connectivity between RDBMSs on different platforms and at different geographical locations. DRDA coordinates communication between systems by defining what must be exchanged and how it must be exchanged. For more details on DRDA, see the *DRDA Planning for Distributed Relational Databases*, SC26-4650-00.

A CICS for AIX application issues an SQL call, using CAE running in the local RISC System/6000 machine. The CAE then uses the services of DDCS to carry the SQL request directly to the distributed data facility (DDF), a component of DB2 for MVS/ESA. The DDF passes on the SQL request to the DB2 for MVS/ESA database server for processing.

Figure 32 shows the products involved when an application running in a CICS for AIX region accesses DB2 for MVS/ESA databases using DDCS for AIX over an SNA connection.

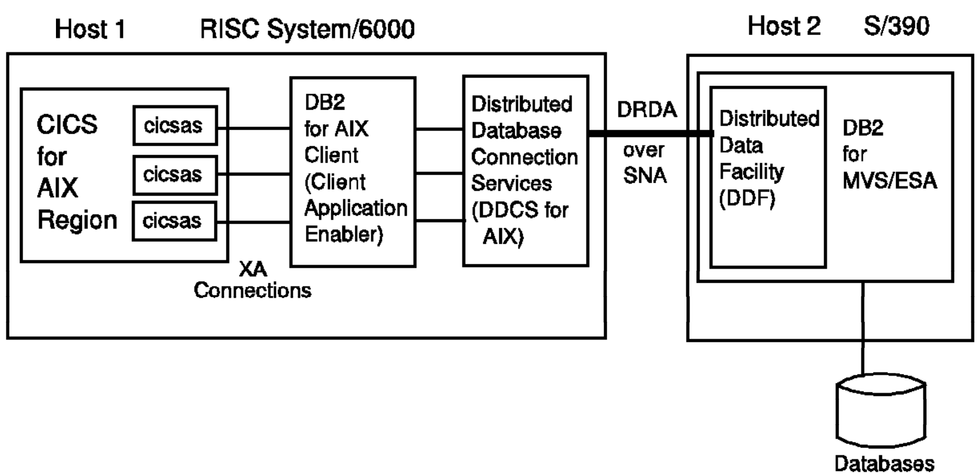

Figure 32. CICS for AIX Access to Remote DB2 for MVS/ESA Database Using DDCS

7.4.3.1 Planning Considerations

A DRDA database located on an MVS/ESA host that is to be accessed using DDCS must be identified by an XAD entry with a unique XA open string. DDCS is XA-compliant and comes with an LU 6.2 syncpoint manager; therefore it can participate in a two-phase commit process, with CICS for AIX as the coordinator.

Configuring DDCS requires detailed knowledge of SNA in addition to knowledge of DB2. Two sets of complex SNA-related definitions with matching attributes on both of the connecting machines must be created. It is very important to ensure that the connection is configured correctly to ensure the proper operation of the remote DRDA database. Information about the attributes must be gathered as part of the planning process for DDCS customization. See the *DDCS User's Guide*, S20H-4793, for more details.

There are security considerations with this type of connection. In the CICS for AIX region, the security policy should be based on CICS security facilities such as transaction security and resource access security for the reasons we discuss in "Security Implications" on page 100.

Security is implemented through the use of authentication and DB2 internal security. Authentication can take place in one or both machines, depending on the configuration of the RDBMSs and the network products. The internal security of the DB2 products then determines what the user can do with regard to DB2 resources.

With DDCS for AIX, it is possible to specify where the user name and password are to be validated when a user tries to connect to a particular database. With your own security requirements in mind, you must plan ahead and decide the location of authentication for each database. The authentication type is specified when you catalog the database in the DB2 system database directory at the local machine. The possible authentication types are:

- AUTHENTICATION = CLIENT

  The user name and password are validated locally. The user is expected to be authenticated at the location where he or she first signs on. Passwords do not flow across the network.
- AUTHENTICATION = SERVER

  The user name and password are validated at the DDCS workstation on which DDCS is installed. Passwords flow to the DDCS workstation over the network.
- AUTHENTICATION = DCS

  The user name and password are validated at the mainframe. Authentication takes place at the host for the user accessing a host database.

The specified authentication type along with the SECURITY parameter value (either SAME or PROGRAM) defined for the APPC node that represents the DRDA connection determine the flow and where validation takes place. This is a complex area. We recommend reading *Distributed Relational Database Cross Platform Connectivity and Application*, SG24-4311.

7.4.4 Accessing a DB2 for MVS/ESA Database through CICS/ESA

Another way for a CICS for AIX application to access a DB2 for MVS/ESA database is to use the CICS intersystem communication (ISC) facilities as the transport mechanism and then access the DB2 for MVS/ESA database from an application running within the CICS/ESA region. Either CICS DPL or DTP can be used to facilitate the communication between the pair of user application programs involved. For simplicity we assume CICS DPL is used.

In the local RISC System/6000 machine, the user application program running in a CICS for AIX region writes the required SQL access information into a COMMAREA and then issues the DPL request (EXEC CICS LINK call). The access information might consist of an SQL select statement along with the associated qualifiers and predicates, for example. The request travels though the Encina PPC Gateway to arrive in the connected CICS/ESA region. In the CICS/ESA region the partner application program is invoked. The application program retrieves the access information from the COMMAREA and builds the necessary SQL call. The SQL call is then issued through the CICS-DB2 interface to DB2 for MVS/ESA server for processing. The result returned by the DB2 for MVS/ESA server can be filtered and further processed by this partner user transaction program before the final result is passed back to the originator of the DPL request in the CICS for AIX region.

The Encina PPC Gateway may or may not be located on a separate RISC System/6000 machine from the machine containing the CICS for AIX region, as required. Both RISC System/6000 machines must belong to the same DCE cell, however. An Encina PPC Gateway is used to facilitate connectivity to the mainframe system over a SNA network while also providing two-phase commit processing. It may be that you already have an existing connection defined to the CICS/ESA region, in which case this simplifies the amount of work to be done, and an existing connection may favor this approach over the use of DDCS, for example, which would have to be set up from scratch.

In this configuration the XA interface is not used for communication with DB2, so there is no need to define an XAD entry. Significant programming effort is required, however, to implement this configuration.

Figure 33 shows how a CICS for AIX application can access a DB2 for MVS/ESA database through the CICS/ESA region.

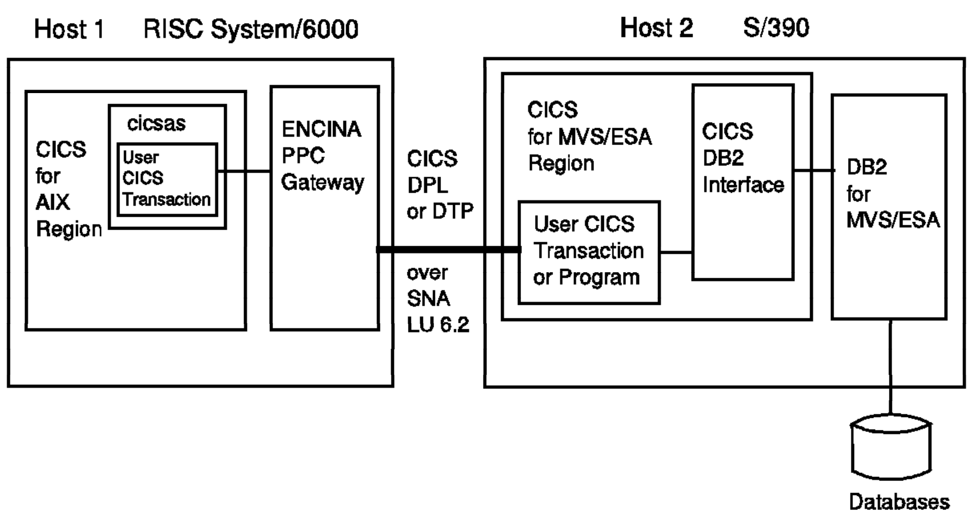

Figure 33. CICS for AIX Access to Remote DB2 for MVS/ESA Database Using CICS ISC

7.4.4.1 Planning Considerations

This configuration is based on the use of the CICS DPL facility. These programs must be developed. There are no special security configuration considerations; normal CICS security facilities apply in this configuration. If resources are being updated in more than one location, that is, one CICS or DB2 system, we strongly recommend that you use two-phase commit support. This support requires the use of an SNA communication product such as SNA Server for AIX. See "Encina PPC Gateway" on page 35 for more details.

As the COMMAREA has an allowable maximum size of 32 KB, its use for storage of the data returned from the DPL request would in turn limit to 32KB the maximum output of the SQL call processed by the DB2 for MVS/ESA server. Where more than 32 KB of data has to be returned, there are two possibilities: filtering of the data returned to the CICS/ESA program to reduce the size of data below 32 KB, or staging the data in a CICS resource such as temporary storage. The data is then returned in 32 KB chunks until it has all been sent. The staging method requires more complex application design and results in more network traffic. Therefore we recommend that you use another of the possible ways of accessing the DB2 for MVS/ESA database from a CICS for AIX region if you anticipate that the size of returned data will exceed 32 KB.

7.4.5 Accessing DB2 for MVS/ESA Database through MQseries Products

Another means of enabling a CICS for AIX application to access a DB2 for MVS/ESA database is through the use of Distributed Queue Management (DQM) along with two different Message Queue (MQ) managers. Use this method instead of using DB2 client/server connectivity as the transport vehicle.

IBM MQSeries products enable applications to communicate through messages and queues. The communicating applications can be local on the same system, or they can be distributed across a wide variety of IBM and non-IBM platforms. A single programming interface, the Message Queueing Interface (MQI), is consistent across all the supported platforms. MQSeries can transfer data with assured delivery. Messages are not lost even in the event of system failures, and there is no duplicate delivery of the same message. The communicating applications do not have to be active at the same time. The sender can still continue doing work without having to wait for the receiver. Message-driven processing is an application style and facilitates the design of applications in discrete functional blocks that are linked by messages. In this way, the units of an application can run on different systems at different times, or they can act in parallel. Faster application development is possible because the MQSeries products shield the network complexity from the application.

In the local RISC System/6000 machine (containing the CICS for AIX region; see Figure 34 on page 107). MQSeries for AIX is installed, and an instance of the MQ manager is started up with two message channel agents (MCAs), one the sender MCA, and the other, the receiver MCA. It is essential for CICS for AIX and MQSeries to be installed on the same system when CICS applications make MQI calls. The MQ manager must be started before the CICS for AIX region is started.

In the remote S/390 machine, MQSeries for MVS/ESA is installed, and an instance of the MQ manager is started up with one sender MCA and one receiver MCA. The DB2 for MVS/ESA database server resides in this same MVS/ESA system.

CICS for AIX and MQSeries for AIX communicate using the X/Open XA interface. This requires an XAD entry to be configured using the switch load file supplied by MQSeries for AIX.

The CICS application running in the CICS for AIX region makes a connection to the local MQ manager using an MQCONN call to the MQI. The CICS application then builds a message with the relevant SQL access information that it intends to pass on to its partner user program in the remote MVS/ESA system. The CICS application then sends the message by issuing an MQPUT call. The local

MQ manager, recognizing that the message is destined for a remote MQ manager, places the message on to the transmission queue. The sender MCA reads the transmission queue and sends the data through the network. The data is transmitted using Message Channel Protocol over a TCP/IP or APPC connection. In the meantime, the CICS transaction can proceed to do other processing. When the message data arrives at the remote machine, the receiver MCA receives the data and puts the message into a target queue.

MQSeries provides a *trigger* facility to start the user application to process the messages when they become available in the queue. With the predefined triggered conditions, the Trigger Monitor program starts up the user program associated with that target queue. The trigger condition could be the receipt of the first message in the particular target queue, for example. The partner user program, which was invoked though the trigger facility, uses an MQGET call to retrieve the message from which it builds the proper SQL call. The partner user program then issues the SQL call directly to the DB2 for MVS/ESA server for processing in the same way that a batch program does. The result returned by the DB2 for MVS/ESA server can be further processed by this partner user program before the final result is passed back to its partner CICS application as a message through the MQ connection in the reverse direction. The CICS application program has to issue an MQGET call to retrieve the message containing the final result.

If you use the CICS adapter of the MQSeries for MVS in the remote MVS/ESA system, you can write the partner user program as a CICS transaction that can communicate with both DB2 for MVS/ESA and MQI. This alternative is less efficient than the direct communication between the CICS for AIX region and the CICS/ESA region.

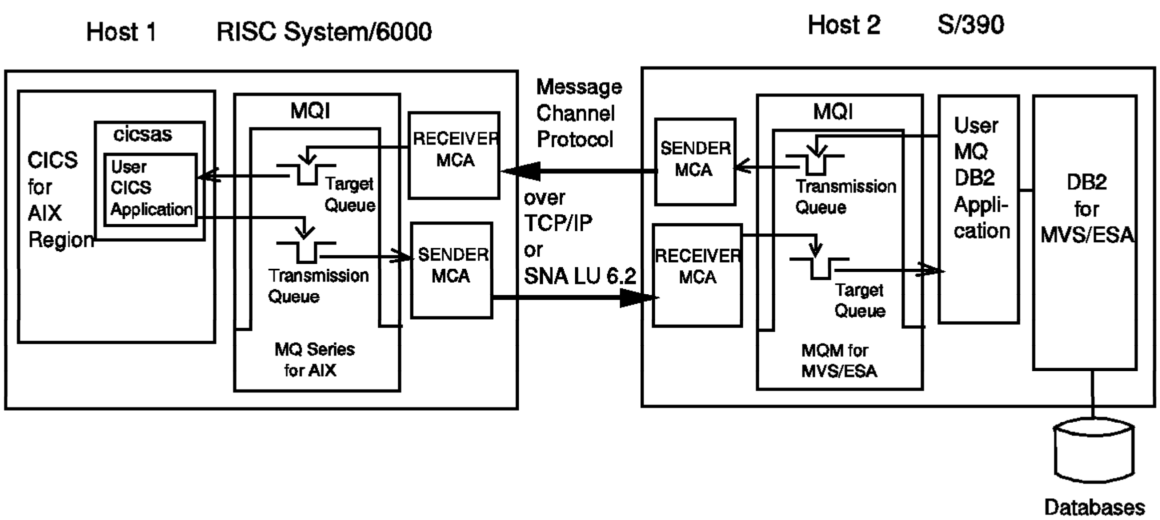

Figure 34. CICS for AIX Access to a Remote DB2 for MVS/ESA Database Using MQ Connections

7.4.5.1 Planning Considerations

The use of MQ connections can make access to DB2 for MVS/ESA databases more complicated than the use of DDCS or CICS/ESA. However, it gives you the advantage of asynchronous message processing and shields you from the network complexity that exists when two machines are not adjacent to each other but connected through a series of intermediate machines. Planning for implementation of the MQ infrastructure is required. Please refer to the appropriate MQSeries manuals for details.

The issues of syncpoint processing and security must be considered.

Syncpoint Processing: Only one MQSeries for AIX MQ manager can be accessed from any CICS for AIX region. MQSeries for AIX supports the XA interface to CICS for AIX, thus ensuring that there is full two-phase commit support. Therefore CICS for AIX can act as a coordinator of transactions in which MQSeries participates. Both MQ messages and other resources within the same unit of work can be committed or rolled back together. The MQPUT and MQGET calls have an option called *syncpoint*. With the syncpoint option, the putting or getting of MQ messages takes effect when the transaction has committed. On the remote MVS/ESA system, there is a nontransactional environment, and so MQSeries provides MQCMIT and MQBACK calls to coordinate MQ messages. It is important to have a good user application design to ensure that MQ messages, DB2 tables, and other CICS resources are properly committed or rolled back in case of any failures.

Security: The security implications that we discuss in "Security Implications" on page 100 relating to DB2 access are also applicable in the use of MQseries products. We recommend that the security policy be based on CICS security facilities. In addition, consider the access control to MQ messages. MQSeries for AIX provides access control for individual message queues. AIX ACLs are used for MQI permissions. The creator of the queue is the owner of the queue, and access to the queue is granted by the owner to other users. In the CICS for AIX environment, the CICS for AIX userid associated with the CICS transaction is used for authorization and access control checking of the MQ resources. On the remote MVS/ESA system, the external security manager, RACF, for example, has to be configured to include the MQ-specific classes of resources for access control and authorization.

7.5 Summary

From a CICS for AIX region it is possible to access a number of different types of RDBMS databases. We focus our discussion on accessing databases that belong to the DB2 family of products. Database access and updates are performed with integrity as a result of using the X/Open XA interface.

CICS for AIX enables database users to make even more effective use of enterprise data distributed over multiple platforms. CICS for AIX enables you to build entire enterprise networks. For example, a desktop application can access data in a remote system using CICS DPL to send access requests to a server application that issues the database calls. These requests may be routed to a server running in AIX and accessing a DB2 for AIX database, or they may be routed onward to a host system running CICS/ESA and DB2 for MVS/ESA. This routing of information can be achieved without the use of any database-specific gateway products.

You can use DB2 for AIX for CICS file management instead of a local or remote Encina SFS. There are some limitations of which you should be aware; the most important one is that nonrecoverable CICS resources mapped to DB2 tables become recoverable due to the inherent nature of DB2 for AIX.

The flexibility and value of CICS for AIX are considerably improved through the ability to access a variety of RDBMS databases.

Chapter 8. Using CICS System Manager for AIX

When an enterprise has a limited number of CICS systems to support its OLTP needs, as the demand for CICS services grows, several barriers become apparent:

- Processing capacity is limited.
- Transaction response times become extended during periods of heavy work.
- With more people working in a few CICS regions, the result of a system outage becomes much more significant to the core business of the enterprise as increased number of users are unable to work.

Such a setup does not best serve the enterprise; in fact it constrains it and could damage it. So, as a result of many pressures, usually business related, the CICS configuration of an enterprise is likely to develop in both size and complexity over time. New CICS systems can be added to:

- Automate more areas of the enterprise
- Add new systems to support new lines of business
- Remodel the existing systems to provide greater availability or improved performance
- Provide a more cost-effective information technology organization
- Develop scalable business solutions so that they can be easily and rapidly expanded should there be a suitable increase in business.

The use of processing technology such as IBM RISC System/6000 SMPs and Scalable POWERParallel (SP) processors can provide significant benefits for the enterprise such as increased application availability and improved performance. The implication of adopting this technology is an increase in the number of CICS regions that have to be managed. An enterprise is willing to pay for the increase, however, because of the gains that the use of the technology brings.

Whatever the reason for the increase in the number of regions to be managed, the problem must be addressed. With such growth it is very likely that systems administration will become more than the current staff can cope with. Employing more people is not the answer. What is needed is a smarter means of working, a way of enabling people to be more productive. Enter CICS System Manager for AIX (CICS SM).

CICS SM was developed to address the management of distributed CICS for AIX Version 2.1 and 2.1.1 regions running on IBM RISC System/6000 systems. The managed environment can range from a small configuration to a large, complex, multitiered client-server configuration.

8.1 Functions

The main system management functions that CICS SM supports are:

- SPOC for definitions, operations, and problem monitoring
- SSI for definitions and operations
- Automated workload management

CICS SM provides a number of additional functions to assist in the systems management of your CICS configuration:

- Discovery by CICS SM of changes to your CICS systems and optional automatic definition of the discovered systems within the CICS SM database. This function can save a significant amount of effort in getting CICS SM initially implemented and assisting with the administration of your CICS systems.
- Event monitoring, which enables you to subscribe to significant events in the life of the CICS systems and their resources. You can subscribe, for example, to the changes in the state of a CICS system. Then, if the health of the CICS system becomes degraded, you are notified and can investigate why there was this change in state.
- A GUI that enables you to act on the objects displayed (and thereby your CICS systems and resources). You select actions through a menu or by dragging and dropping.

Let us look at the main CICS SM functions in a little more detail.

With SPOC it is possible to operate all CICS systems and resources in the enterprise from one workstation session. The physical location of the machine on which these resources are located does not have to be known by the person managing them. You log on once to CICS SM and from there manage all CICS systems and resources without having to know where they are.

An SSI greatly reduces the impact of growing complexity. A user interacts with one entity rather than a complex collection of CICS systems and resources. A command can act on all CICS systems and resources in a group. It also reduces the risk of failure of the operation. When manual changes are made on many systems, one of the changes could be incorrect and have a potentially serious impact on the CICS systems and hence the business of the enterprise.

You can use one action on a CICS system model to act on a group of CICS systems. A CICS system model defines the initial values of attributes for one or more identical systems. Similarly you can use one action on a configuration to affect all CICS systems in the configuration. You could, for example, activate a configuration to make all CICS systems available on several AIX nodes. One command can operate on all of the necessary CICS systems and resources. The same command does not have to be repetitively issued on each of the systems.

8.2 Workload Management

CICS SM workload management (workload management) with a suitable CICS configuration can be used to improve the availability of applications in the event of a system failure, make better use of the available processing power, and allow seamless maintenance to be carried out. It is the function of workload management to determine which CICS system is most appropriate to process a piece of work.

Any CICS for AIX client or region with a program to run can be told which is the best system to route work to on the basis of:

- Workload separation requirements, that is, the need to route particular transactions to a particular group of CICS systems according to user-supplied separation rules
- Workload balancing requirements, that is, the need to balance the workload across a group of CICS systems with the decision as to which system to

route the work to taking into account the availability and activity levels of the systems
- The availability of required resources
- The performance and health of the CICS systems

Workload management can manage work requests from the following sources:

- Dynamic transaction routing
- Distributed program linking
- Routing of CICS Client for AIX ECI requests
- Session-level load balancing for CICS Client for AIX EPI sessions
- Session-level load balancing for the CICS 3270 emulator, cicsterm
- Session-level load balancing for CICS Telnet server (cicsteld) clients
- Data-dependent workload management

CICS SM uses object-oriented technology and relies on System Object Model (SOM)/Distributed System Object Model (DSOM) and Semantic Network for key functions. Each item to be managed within CICS SM and each entity displayed by the GUI is represented by an object. Think of an object as a tangible representation of something you can manage. For example, there are classes of objects for CICS systems and CICS resources such as connections, files, programs, and transactions.

CICS SM has the concept of a management node and managed nodes. The management node is the node in the complex from which the SPOC and SSI are performed. The managed nodes are those nodes that have CICS for AIX systems that will be placed under the control of CICS SM.

Note: It is important to understand exactly what CICS SM V1.1 does and does not manage. It facilitates systems management of CICS for AIX regions and provides workload management for CICS for AIX regions and clients. It does not manage any other CICS on Open Systems regions. It does not allow any other CICS on Open Systems clients or IBM CICS clients to be workload managed directly (the work they generate could be managed if the clients are attached to a CICS for AIX region). Work to IBM Mainframe CICS systems cannot be managed either.

8.3 Some Terminology

Within the CICS SM publications, an instance of a CICS for AIX server is referred to as a *CICS system* The CICS for AIX publications refer to an instance of a CICS for AIX server as a *CICS region*. *CICS system* is synonymous with *CICS region*. The two phrases are interchanged freely within this book.

CICS SM holds in an internal database definitions of all the CICS systems and resources that you choose to manage from within it. These definitions make it possible to re-create a CICS system multiple times. Many of the terms within CICS SM are based on the need to formalize the definitions of a CICS system and resources coupled with the fact that CICS SM is implemented with object-oriented technology.

In an environment in which there is no CICS SM, the only way to easily create a CICS system and configure all of its resources is to export the region and reimport it each time or develop some scripts to do the work. This does not make it easy to create another similar but slightly different region. You would have to create the region and then modify it. With CICS SM it is possible to hold the definitions for many different types of CICS systems or resources, easily define the precise CICS system that is required, and then create it. Even when the region has been destroyed, you can still re-create it even if you did not export it first because CICS SM has a representation of what that CICS system looked like. The terms and definitions of objects within CICS SM are geared to creating this definitional view of the world.

Each CICS system to which you want to route work is defined by a *system clone*. To create the system clones you define a *system model*, which specifies the attributes common to the clones. There might be one system clone for production systems, and another for test systems. Alternatively there might be a clone for the system that runs the payroll system, and another for the ledger system. The contents of the clones can be whatever is meaningful to your installation. The system model undergoes the process of cloning to produce a system clone. It is possible to clone a system model many times. Each system clone represents a unique instance of the system model.

The concept of a model and clones also applies to the resources within CICS for AIX, such as transactions or gateways. For example, there are *transaction models* and *transaction clones*. To have an instance of a resource model, such as a transaction, the resource model must be configured onto the system model. When the system model is cloned, appropriate resource clones are created automatically on the system clones, thus creating the mapping (needed for workload management) between the resource that a user requests and the systems on which it is available.

A group of system models and resource models and their associated system clones and resource clones are known as a *configuration*. Each configuration represents a possible implementation of a *Plex*, a named set of configurations. It consists of CICS systems and resources with particular parameter settings. So, for example, there might be one configuration for day and another for night. Only one configuration within a Plex can be the target, that is, the group of systems and resources with the characteristics that you intend to be up and running. There might be a Plex covering the payroll system or the London installation. This Plex would have the possible configurations within it. So, for example, in our London Plex we might have weekday and weekend configurations. The difference between the two configurations is that the weekend configuration does not have the warehouse system active, only the customer query system.

As we may want to share system or resource models between different configurations, CICS SM has an object called a *template*, which is the highest level of abstraction within a Plex. There are system templates and resource templates. It is possible to create a system template from a system model. It is then possible to create a new model based on that template in another configuration. A similar process applies with resource models.

For information on the terms used within CICS SM see *IBM CICS System Manager for AIX Product Overview*, GC33-1592-00.

8.4 Components

In this section we provide an overview of the major components of CICS SM and explain the roles they play so that you can configure CICS SM for the maximum value in your environment.

Figure 35 shows the major components of CICS SM.

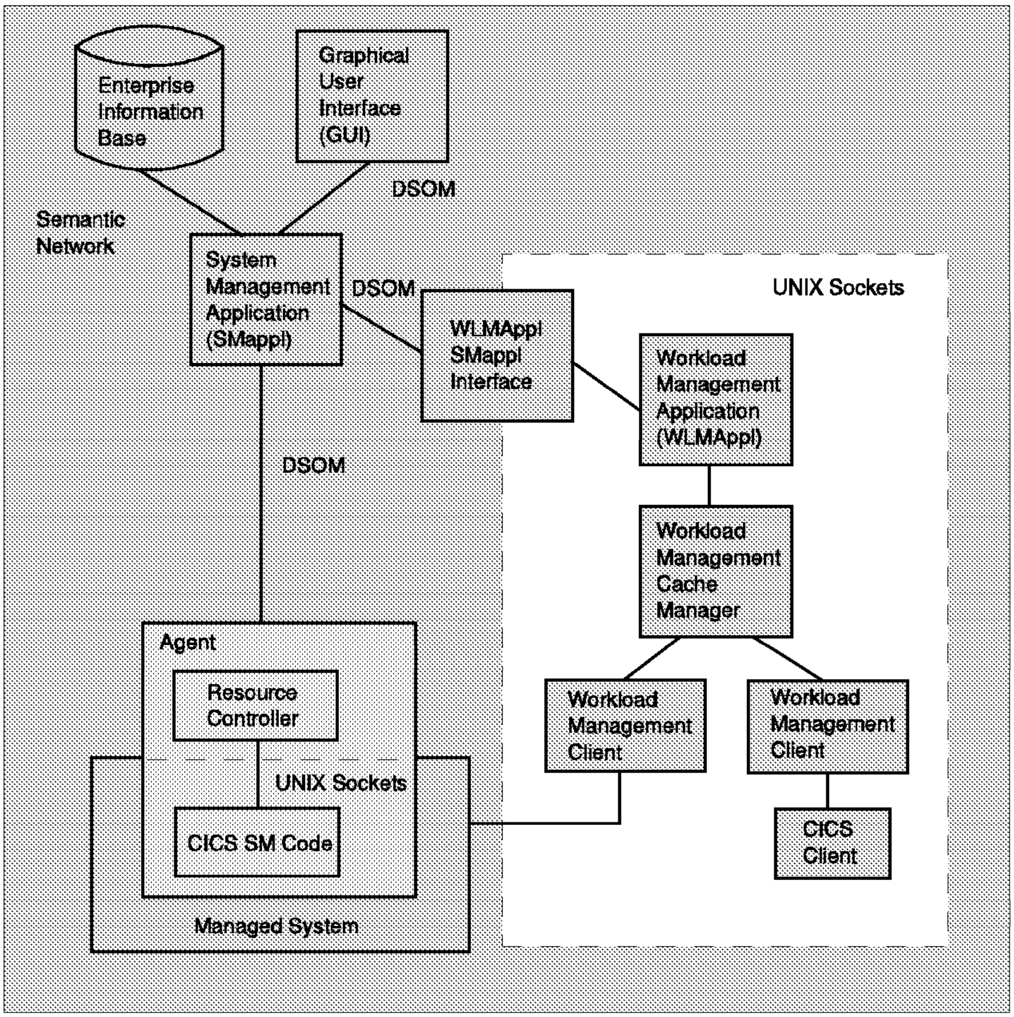

Figure 35. Major Components of CICS SM

8.4.1 Enterprise Information Base

The enterprise information base (EIB) is the long-term data storage used by CICS SM. Each system management application (SMappl) has its own EIB, which it stores as an instance network file. The instance network file is a database for CICS SM definitions. These definitions represent the CICS systems and resources that are to be managed. See *IBM CICS System Manager for AIX Product Overview*, GC33-1592-00, for more information about an instance network and the definitions held within it.

8.4.2 System Management Application

The SMappl provides the logic that manages part or all of an enterprise. It relates the definitions stored in the EIB to the real images of systems and resources in the real enterprise, that is, the actual definitions in the systems being managed. The SMappl composes the data defined to CICS SM with the data that it can determine from the real world. The composed data is presented to you on the GUI. It is through the GUI that you operate systems and resources on the managed AIX nodes.

8.4.3 The Agent

The CICS SM agent acts as the delegate of the SMappl on a managed node. The agent interacts directly with the managed objects, passing data back to the SMappl and acting on the managed objects when directed by the SMappl. The agent comprises a resource controller and some system-specific code. One resource controller runs on each AIX node that has CICS systems to be managed. It runs independently of those systems. A single resource controller typically can connect to multiple managed systems on the same node. Some operations cannot be directed to a CICS system. Operations such as the install, start, and uninstall of a CICS system are handled by interactions with the operating system.

Within each CICS for AIX system to be managed there is a long-running CICS task. This task contains CICS SM code. It enables communication with the systems resource controller. The long-running transaction is started when CICS system initialization is started. It establishes communication with the resource controller to which the managed system is connected. Requests from the resource controller to the managed system along with notifications from the managed system to the resource controller are sent by means of the Internet Protocol transport mechanism.

8.4.4 Workload Management

The workload management component consists of the following:

- The workload management application
- The workload management cache manager
- The workload management client

8.4.4.1 Workload Management Application

Workload management makes routing decisions on the basis of data that it has available to it about the CICS systems and resources. This data is held in two types of cache: global and local. Both caches are similar in structure and are based on a semantic network. There is one global cache and potentially many local caches. The global cache is used by the workload management application (WLMAppl). A local cache is used by the workload management cache manager and a workload management client (see 8.4.4.2, “Workload Management Cache Manager” on page 117 for more details).

The WLMAppl adds information about the currently active systems to a global cache that it has discovered from the SMappl about the currently active systems. This information is obtained at WLMAppl startup. Only sufficient information for WLMAppl to manage the work of each CICS system is passed to it. With this global cache, WLMAppl can resolve workload routing requests faster than if the EIB had to be accessed each time.

The WLMAppl subscribes to changes in state of managed objects, that is, a CICS system or resource, so that it can keep the global cache up to date. When the WLMAppl is subsequently informed by the SMappl of changes, it updates the global cache. The WLMAppl is also responsible for informing the workload management cache manager of relevant changes made in the global cache. Thus the workload management component is kept abreast with what is happening in the real world. For example, if a transaction is no longer active in a particular CICS system, the cache must be updated so that routing decisions for new pieces of work can take into account the fact that the transaction is no longer available in that CICS system.

8.4.4.2 Workload Management Cache Manager

There is typically one workload management cache manager on each system on which routing decisions are to be made.

The workload management cache manager interfaces with the WLMAppl to transfer data from the global cache to the local cache. It also processes requests for data from workload management clients. If the requested data is not available in the local cache, a request is forwarded to the WLMAppl to get the data from the global cache. If the data is not available in the global cache, the WLMAppl requests it from the SMappl. The workload management cache manager also informs clients that their requests have been satisfied or that the requested data could not be found.

8.4.4.3 Workload Management Client

There is one workload management client for each CICS component that wants to determine which is the best CICS system to which to route work. CICS SM provides a CICS exit for dynamic transaction routing, DPL, CICS Client for AIX ECI work, CICS Client for AIX EPI work, and data-dependent routing. For session-level load balancing, CICS SM provides versions of cicsterm and cicsteld called *bhgwterm* and *bhgwteld*, respectively.

The workload management client performs the following functions:

- Processes requests from CICS clients that want to determine to which CICS system to route work
- Searches the local cache for the data to fulfill the workload management requests from CICS clients
- Applies a workload balancing algorithm to determine which of a set of CICS systems is best able to do the work for a CICS client and then updates the cache to reflect the decision that it has made
- Requests information from the workload management cache manager when it cannot be found in the local cache

8.4.5 Graphical User Interface

The CICS SM GUI is based on a set of windows that have standard elements common to many GUIs, such as a title bar, maximize and minimize buttons, a tool bar, and a status line. The functions of the GUI are based on a menu structure that can be accessed through the menu bar items and their pull-down menus. CICS SM has been designed so that it can be operated from the menu bar only; that is, without the tool or action bars. This is particularly useful for users with limited screen space. Figure 36 on page 118 shows the layout of the CICS SM GUI.

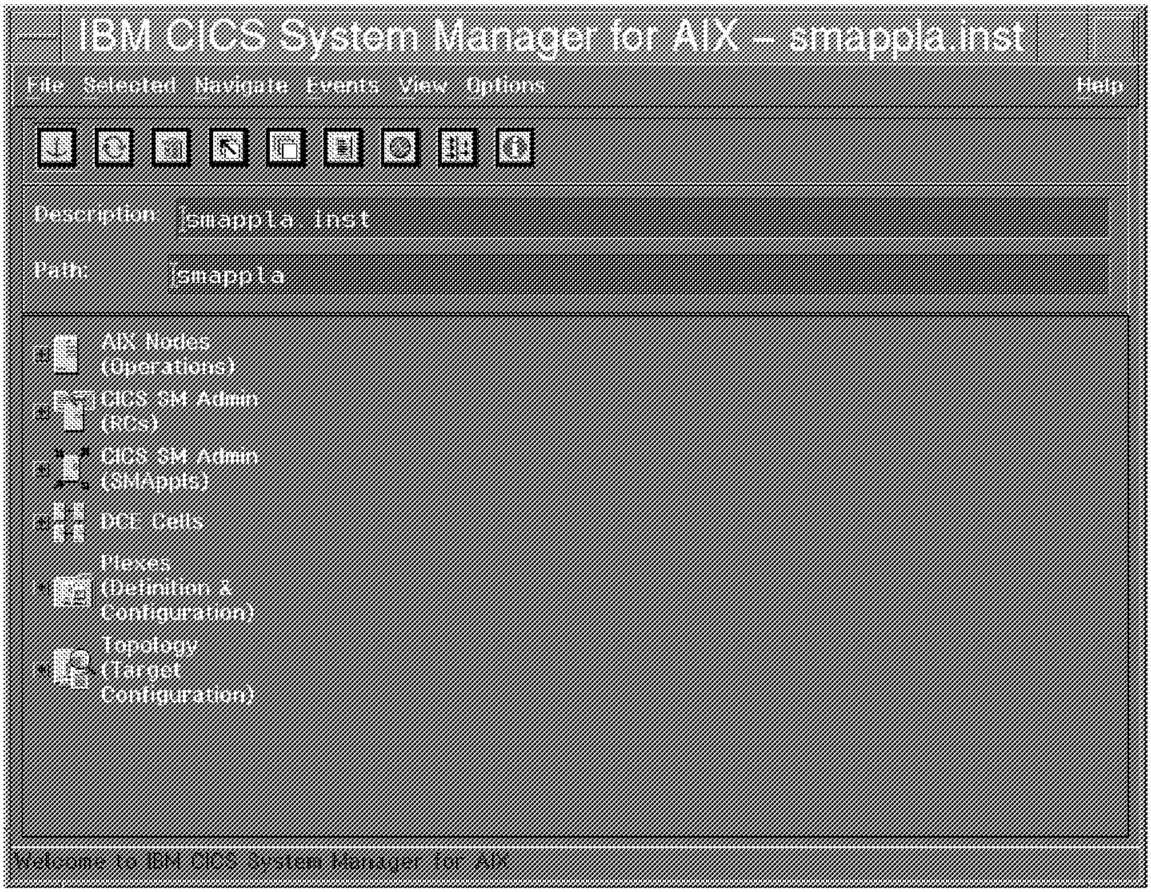

Figure 36. The CICS SM GUI

8.4.6 Distribution of Components

You must decide where to locate the CICS SM components. To assist you in the decision, we discuss the types of intercomponent communication that take place within CICS SM. We then make our recommendations for component distribution.

8.4.6.1 Intercomponent Communication

Two types of intercomponent communication are used within CICS SM. They are DSOM and UNIX sockets. The use of these communication methods enables distribution of CICS SM components across multiple AIX systems. DSOM is used between those components that have a need for its services. UNIX sockets are used within the workload management components where high performance is essential. DSOM does not have to be installed on every machine, only the management node and the managed nodes. In Figure 35 on page 115 you can see that intercommunication between the SMappl and the WLMAppl-SMappl interface, the resource controller, and the GUI uses DSOM. Intercommunication between the WLMappl and the workload management cache managers uses UNIX sockets with a datagram protocol, as does the long-running transaction in its communication with the resource controller. Workload management client to CICS client communication occurs as a CICS exit call. There is no explicit communication as there is with DSOM.

8.4.6.2 Considerations and Recommendations

Although the use of DSOM and UNIX sockets allows some freedom in deciding where the major components are located, there are some considerations to be borne in mind before implementation:

- For those components that require DSOM to communicate with other components, DSOM must be installed on the same system. Otherwise the component will not be able to contact the other components.
- The SMappl, WLMAppl, and GUI typically form a management unit and reside on the same AIX machine. There may be several instances of such management units, each residing on differing AIX nodes and each forming a separate management unit.
- Although it is possible to place each of the management components on a separate machine, it would not be particularly wise to do so as this would increase the amount of intermachine traffic, which is less efficient than intramachine communication.
- Each SMappl has its own EIB. The SMappl is linked to one or more resource controllers. A resource controller is required on each node on which there are CICS systems to be managed.
- The global cache used by the WLMAppl is held on the same system as the WLMAppl. In a large configuration there would typically be multiple workload management cache managers, that is, one on each system on which there is work to be managed.

Figure 37 on page 120 illustrates the recommendations that we have just covered. You can see that the figure contains four different nodes covering three different node types:

- Management node
- Managed node
- Client node

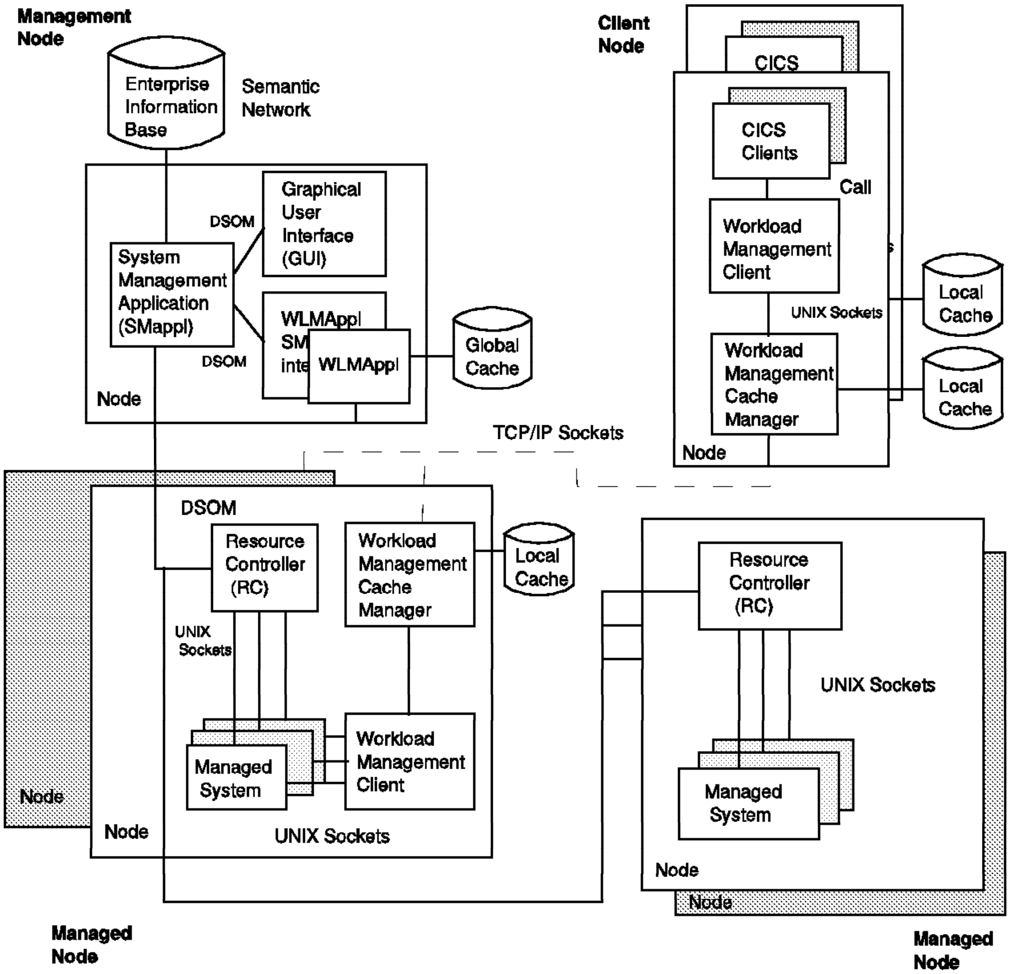

Figure 37. CICS SM Component Distribution

Management node: The GUI, SMappl, WLMAppl-SMappl, global cache, and EIB are located on the dedicated management node. DSOM is installed on this node to facilitate communication between the GUI and the SMappl as well as between the WLMAppl and the SMappl.

Managed node: Each of the managed nodes contains systems to be managed, DSOM, and a resource controller. Additionally, one of the managed nodes contains a workload cache manager and a workload management client. These workload management components are present because one of the managed systems is using workload management for dynamic transaction routing and DPL requests to the other managed systems. The management node that does not have the workload management components installed has no need to perform workload management.

DSOM is installed on the managed nodes to facilitate communication between the resource controllers and the SMappl. The resource controller communicates with the agent transaction in the managed system through UNIX sockets and a datagram protocol. The workload management components communicate with each other through UNIX sockets and a datagram protocol. Communication between the workload cache manager and the WLMAppl on the management node is also achieved through UNIX sockets and a datagram protocol over TCP/IP as the communication is now intermachine.

Client node: The client node contains a CICS client, a workload management client, a local cache, and a workload management cache manager.

DSOM is not installed on the client node as there is no need to communicate with the other CICS SM components that use DSOM, namely, the GUI, SMappl, and WLMAppl-SMappl interface.

The workload management client is called by the CICS client to route the client request. It does this based on information held in the local cache. The workload management cache manager manages the local cache, updating it when updates are received from the WLMAppl or requesting data from the WLMAppl when a routing request is made for a program or transaction that is not present in the local cache.

8.5 Workload Management Considerations

In this section we look more closely at three aspects of workload management. First we examine the sort of configurations that are required to improve the value that can be obtained from introducing workload management. Second we look the reference system that is used with workload management in order to rank the size of different processors in a CICS configuration. Third we look at the workload management algorithm.

8.5.1 Maximizing the Benefit

Workload management is most effective when there are a number of CICS systems from which to choose. There is little point in implementing workload management with a single CICS system, unless you are planning to subsequently expand, as there is only the one system to which to route. Static routing would be just as effective in such a case.

8.5.1.1 Combining Resources

The ideal approach for workload management is to have one (or more) applications available with each CICS system on its own machine. It may be that you choose to adopt this approach for only your most critical applications. There could be significant value, however, in combining several less obvious candidates for workload management and the machines on which they run. The aim is to provide one common type of CICS system that contains all of the resources required to run each of the applications. This common CICS system is then reproduced on multiple systems. Workload management is used to distribute the incoming work over the systems that can process the work. Such an approach can provide benefits to each of the applications as the combined machine resources are now available for the applications. Such an environment cannot proceed without the use of effective workload management. The availability of CICS SM workload management now enables you to adapt in the distributed transaction processing world.

Consider the case shown in Figure 38 on page 122.

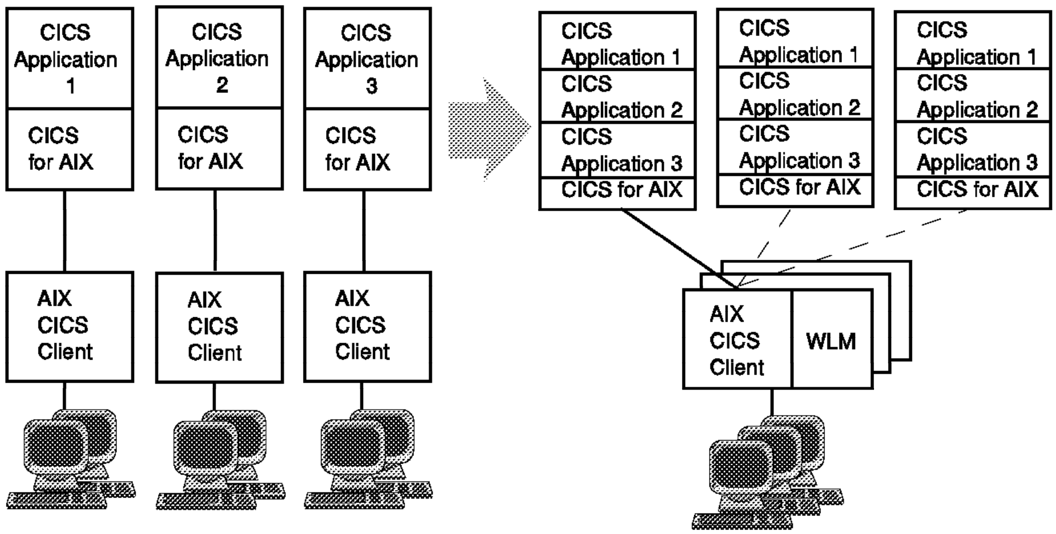

Figure 38. Combining Resources

In this example we have three separate systems, each with a CICS for AIX system running a separate application. The application on machine 1 is approaching the limit of its CPU capacity. Machines 2 and 3 are lightly used. By restructuring the environment so that each application is available within each CICS system, you can provide some significant improvements to the service that users receive when workload management is added to the restructured environment. Now, each of the three applications is available in each of the CICS for AIX systems. Thus the work of each of the applications can be split across the processing power of the three systems. This is most significant for application 1, as it was approaching the CPU limit of the machine on which it was running. There is now additional capacity available to it, so it is not necessary to buy a larger machine for running the application. Applications 2 and 3 have also gained. They have greater availability as they are available on more systems. In the event of a failure of any of the CICS systems or machines, all of the applications will still be available to their respective users. The performance may not be as good as it was before the failures, but the applications are available, which is the most important consideration.

8.5.1.2 Multiple CICS Systems on One Machine

Although the ideal approach is to be able to combine machines to provide additional resilience to a workload managed environment, benefit can also be obtained by providing multiple CICS systems on a single machine. This approach gives greater availability in the event of a CICS system failure. The gain is less than it would be with multiple machines, as the work is supported only on a single machine. An outage of that machine will take both systems down. Multiple regions on a single machine is not particularly recommended unless there is the machine processing capacity to cope with it, ideally an SMP. Adopting such an approach on a uniprocessor would have questionable value as there would be two regions competing for CPU where previously there had been one.

8.5.1.3 Using an SP Processor

The use of an SP processor is a good example of an environment where CICS SM provides significant value to a CICS configuration. A highly available, flexible, and easily serviceable configuration can be built. Figure 39 shows the logical view of such an environment.

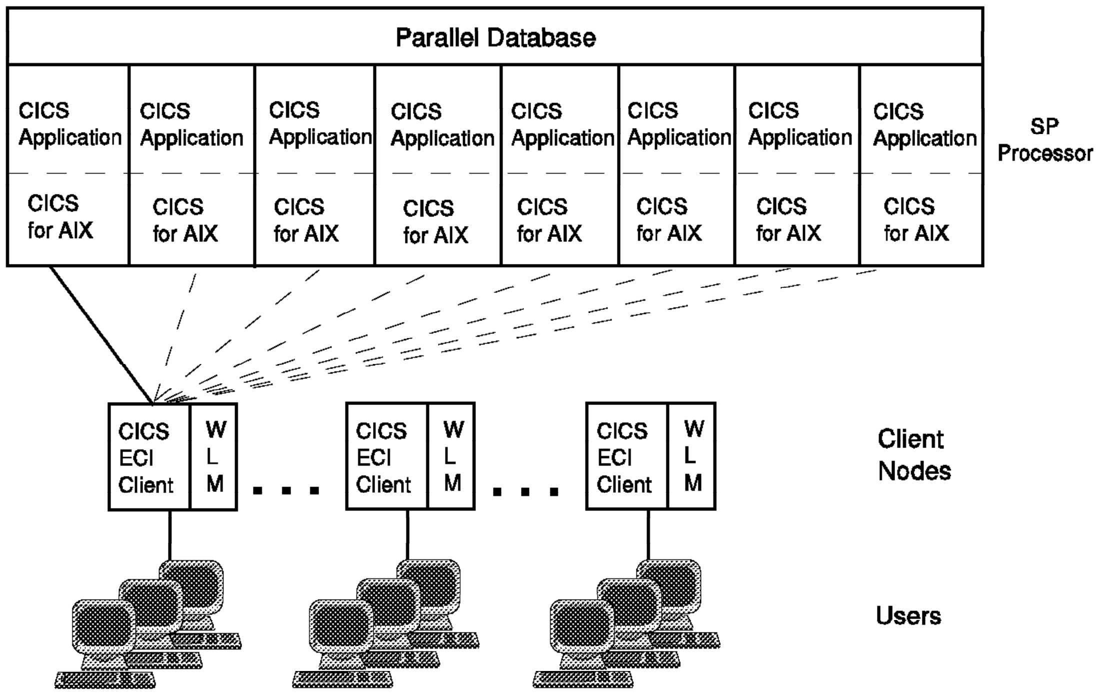

Figure 39. CICS on an SP Processor

A CICS for AIX region exists on each of the nodes of the SP processor. A parallel database, such as DB2 Parallel Edition, is used to hold the application data. The application is available within each of the CICS for AIX systems. Each of the CICS for AIX systems is connected to DB2 Parallel Edition. Incoming pieces of work are routed among the available CICS regions by workload management. This configuration provides multiple nodes, each capable of accessing any of the data. In the event of a node or CICS system outage, the other nodes will continue to process the incoming work. With CICS SM, it is possible to shut down individual CICS systems and apply maintenance or housekeeping procedures while the rest of the complex continues to process work. When the maintenance is complete, the CICS system can be restarted and reintroduced to play its active part in processing the user′s work requests.

8.5.1.4 Application Considerations

In combining applications into a single CICS system type, you must ensure that applications:

- Can access the data required

  If multiple regions are to be capable of running an application, each region requires access to the data used by that application. If the data is held in a serial database manager, you may want to consider accessing that database remotely; that is, each CICS system will have an access path to the data. Be aware that accessing data in this manner has an impact on performance. There is no problem with a parallel database, as each CICS system would have access to the database. Where the data is held in an Encina SFS, to provide universal access to the data, each CICS system would have to access the data remotely. This is quite easily done as long as both the CICS system and Encina SFS are in the same DCE cell. Again though, there is a performance overhead in accessing an Encina SFS remotely. The most immediate by-product is that fast local transport cannot be used between the CICS application server and the Encina SFS.

- Have all of the facilities required

  In combining applications in the way in which we have suggested, you must remember to include all of the facilities that the applications require. For example, you may have to define more intersystem communication links because the applications being placed within a region each access different remote systems.

- Are compatible

  Different applications may try to use the same temporary storage or transient data queue names, for example. If the names cannot be changed, the conflicting applications would have to remain separate.

- Do not contain any affinities

  An application contains an affinity when, for example, it must be run in the same CICS system because a piece of data or state information that it requires is in that CICS system. From a workload management point of view, this is a bad characteristic for an application because it restricts the flexibility of making routing decisions.

For a fuller description of the problems that you may encounter in running in this type of environment, see *CICS/ESA Dynamic Transaction Routing in a CICSplex*, SC33-1012-00. That book was written with a parallel transaction server environment in mind but contains useful information that is applicable in a parallel environment.

8.5.1.5 Summary

You may find that moving to the multiple CICS systems we suggest here requires significant effort. Indeed it may not be achievable immediately, given the characteristics of some applications. It is a worthwhile goal to achieve, however, for a number of reasons:

- Increased application availability
- Better utilization of the available resources
- Good growth capability. As the demands grow on the system, it is possible to produce another instance of the common system and have it added to the workload management
- Transparent maintenance

8.5.2 The Reference System

CICS SM workload management recognizes that machines have different processing capabilities. If one machine contains a faster processor than another, all else being equal, the first machine will process more work in a given period of time.

To deal with these differences in processing capability, workload management uses a reference system. In an environment in which there are mixed processor speeds, it is possible to rank machines relative to each other. For example, the middle speed machine in your configuration might be given a relative speed of 1. Those machines that are slower would have a value of less than 1, and those that are faster would have a value of more than 1. If all machines were of identical speed, they would all have the same relative speed values coded. A value of 1 would be sensible.

It is important to set these values appropriately, particularly if there are machines of different speeds in your workload management configuration. The default is for all systems to be treated equally; that is, a relative speed of 1.

To set the relative speed of a system you can edit the appropriate system model or system clone and specify a value for the speed of the system relative to the reference system. If you have a system model for each type of processor, set the value at the system model level. Otherwise you will have to set it at the system clone level. If the CICS system is subsequently moved to another machine with a different relative speed, you would have to amend the value accordingly.

8.5.3 The Algorithm

In this section we take a closer look at the workload management algorithm for selecting a system to which to route a piece of work. We discuss the principles of the algorithm rather than the specific parameters. For a description of the workload management parameters, see *IBM CICS System Manager for AIX User's Guide*, SC33-1594-00.

The overall effect of the workload management algorithm is for clients to choose a system clone for each work item based on the best information available. The choice is based on a simulation of information not available locally and historical data gathered by the clients themselves.

There are two aspects to choosing a system clone on which to run an item of work:

- Identifying the system clones that have the required application, transaction, or program
- Choosing the system clone that provides the fastest service

The workload management client identifies system clones that have the required resource from information in the local cache. It calculates a score for each system clone and selects the clone with the lowest score. If several system clones score equally, the algorithm selects one at random. This randomness ensures that work is spread evenly initially or after a quiet period.

The total score of a system clone is the sum of scores representing the load already on that system, its health, the quality of the connection to the selector, its performance history, and the extra load that deciding to route the current work item to this system would cause:

```
total_score = load_score
              + health_score
                + connection_score
                  + history_score
                    + projected_new_load_score
```

The calculation of each of the above terms is dependent on parameters that are attributes of the requested resource model, the system clone being considered, its model, and the active configuration. These parameters are not discussed at any level of detail here.

8.5.3.1 Load Scoring

The load_score value is a measure of the outstanding level of work that remains to be completed by a system clone. When a system clone is selected, its load is increased to reflect that an additional piece of work has been routed to it. The size of the load that is added is equal to the program or transaction "Execution time on a reference system" divided by the number of application servers in the clone to which it has been routed. When the work item is complete, the response is detected by the workload management client, and the system clone′s load is decreased by the amount that was added when the system was selected.

The workload management client code runs a simulation to predict how busy its candidate systems are. This simulation is preferred to an actual measurement of how busy they are; as with the time delays involved, such a system would not form a stable servo loop. The overall effect of the simulation is to discourage the selection of systems that have large numbers of sessions or recently have been given large amounts of work.

The workload management algorithm acts on static data provided by CICS SM together with dynamic information it has obtained by remembering the routing decisions it has made and observing the responses. Note that the dynamic information is obtained from all of the processes that use one workload management cache manager and therefore run on the same node as the workload management cache manager. There is no sharing of dynamic information between workload management cache managers: each cache provides information that its clients use to make decisions independent of clients on another node. As a consequence, nodes running multiple workload management client processes have a larger pool of dynamic information on which to balance the workload than nodes running one or very few workload management client processes.

8.5.3.2 Health Scoring

The health_score value is determined by taking the health of a system clone (as supplied by the health monitor) and applying an adjustment value corresponding to the health of the region. Health values range from 1 (excellent) to 5 (very poor). The use of these health values discourages the selection of system clones that the health monitor reports as inferior.

8.5.3.3 Connection Scoring

The connection_score value is typically constant for a given system. It is determined by multiplying the reference time for a round trip on the connection by the connection delay weighting. The value can change only when a new configuration is activated.

8.5.3.4 History Scoring

The history_score value is calculated according to which of these three different situations applies:

- If the client has not selected the system clone before, the history_score value is zero.
- If the client has selected the system clone before, but not on the last occasion, the history_score value is equal to the value of the configuration attribute: *Bias towards previously used system.*

- If the client selected the system clone on the last occasion, the history_score value is equal to the value of the configuration attribute: *Same system bias*.

The overall effect of the history_score value is that, after a system clone has been used, its client has a bias toward continuing to use it. It is important to set the *Same system bias* to a value greater than that specified for the *Bias toward a previously used system* for a configuration. The intention is to provide a bias toward any system used before, but particularly toward the system last used. This will only occur if the *Same system bias* is greater than the *Bias towards previously used system*.

Clients can remember a limited number of system clones. If a previously unselected system is selected, the least recently used is discarded if necessary to keep the number of known system clones to the predetermined limit. The assumption here is that, after a time, the advantage of using a system clone that has previously been used disappears (for example, the terminal has been discarded, or files are no longer cached).

8.5.3.5 Projected New Load Scoring

The projected_new_load_score value is intended to represent the amount of additional time a system would be busy if it were selected to run the current work item. Hence it is the projected load, were this system to be selected. If the system is selected, this amount is added to the load score and taken into account in future decisions. However, if the system is not selected, this amount clearly does not get added to the load score. By including this component to the calculation, we can take into account differences between systems and select a system that gives the fastest response time. For example, if two systems had all other scores equal, the projected new load score would cause the selection of the system with the greater power (that is, system speed relative to the reference system).

Another important use of this element is specific to the ECI case, where there is known to be a large overhead when a terminal has to be installed to run a transaction. The load associated with an autoinstall is an attribute of the configuration (*load due to ECI autoinstall*). When, in the ECI case, the workload management algorithm can predict that an autoinstall would be needed to route a work item to a particular system, the algorithm increments the projected new load score by the *load due to ECI autoinstall*. This heavily biases against systems where a terminal installation would be needed to run the transaction.

8.5.3.6 Score limiting

System clones with a total_score greater than the maximum score permitted are rejected as unsuitable. The default settings ensure that all system clones with a health of 5 will be rejected. The limiting is important only if all potential system clones are in this state. In this case, an alternative partition or the CICS default system is used.

8.6 Planning Considerations for CICS SM

Before implementing CICS SM in your environment, it is important that you understand what CICS SM has to offer. You must be able to distinguish between the roles of systems management and workload management. You must also be clear about which components of CICS SM are used in which roles. As with the introduction of other software components, it is important to put a lot of thought and effort into the early planning stages. If the role of systems management itself is not new in your installation, the technology that CICS SM uses almost certainly will be. Starting small rather than going for a complete change at once will enable you to become more familiar with the product.

The emphasis in the sections that follow is on planning your configuration and ensuring that it meets your requirements.

8.6.1 Overall Objectives

Be clear about what it is that you want to gain from the introduction of CICS SM into your environment. Are you looking to provide systems management or do you want workload management active as well? Consider whether you are prepared to introduce change into your existing environment to achieve your objectives. Introducing systems management into an existing configuration is relatively easily done and does not necessarily require change. Implementing workload management is a somewhat more involved task, and depending on your current configuration, it may well require change in order to obtain the best value from it.

8.6.2 Understand Today′s Configuration

If you are going to remodel your existing CICS systems to combine applications into a few multipurpose application-serving regions, most of the work is CICS related. CICS SM is the catalyst for the change. You will most likely have your own procedures for remodeling CICS systems. Below are some issues to address in the planning and execution phases of the remodel.

- Which applications do the CICS systems support?

  Ensure that you have a thorough list of the applications that are running within the CICS systems that are to be remodeled.
- Which CICS facilities are required to support those applications?

  Remember that these multipurpose application-serving regions will be the sum of the different types of regions that existed. Some applications will use facilities or resources that others do not. All should be included.
- Are there any special requirements for those applications?

  An application might have special operational requirements, such as having to be available for less time in the day than other applications. This is not a problem with CICS SM as it can stop some applications while leaving others active. The key thing is to be aware of all of the different requirements.
- Where is the data held for each application?

  Ensure that all applications have access to the data that they require regardless of where it is held.
- Are there any application affinities?

  The most preferable solution to dealing with application affinities is to remove the affinities, but this is not always feasible. Alternative approaches are to use workload separation or run the applications in their own CICS systems.

- Is the enterprise business structure represented?

  CICS systems and applications exist to serve the enterprise. If the CICS systems have grown apart from the enterprise business structure, perhaps because of changes within the enterprise or takeover and merge activity, now is a good opportunity to make the necessary changes to update the configuration.

- How does work arrive to be processed by the CICS systems?

  You must understand how work arrives to be processed by the CICS systems. Does it arrive from another system or is it from one of the CICS clients? Remember that it is only possible to workload manage AIX CICS clients directly. With the IBM CICS Clients you would have to have the clients attach to a CICS for AIX region, which then manages the routing of work. This is not an ideal solution because of the overhead that going through a CICS for AIX system adds. The solution does allow work to be workload managed to your CICS for AIX systems, however, when it would not otherwise be possible.

- Are all processors equal?

  Know the relative processor speeds of the machines that will have work routed to them. This information must be entered for each system clone that is being workload managed. See 8.5.2, "The Reference System" on page 124 for more details.

- What is the current performance?

  Before introducing CICS SM, collect data that enables you to understand the current performance of your systems. This data should include transaction rates, user response times, and processor utilization. Such data can be invaluable for comparison purposes after the introduction of CICS SM. The response time data is required to specify the *estimated execution time on the reference system* for both transaction and program models. The data is also helpful for pinpointing any problems that you may initially experience.

8.6.3 New Configuration Considerations

If the structure of the configuration is to be changed, the transition is likely to be largely controlled by application migration considerations.

To produce a new logical and physical organization showing the components in the system, where work arrives from, where it is processed, and how the routing decisions are made, you will want to look at:

- Location of the systems management function

  Decide where the management focal point will be. We recommend that the GUI, SMappl, and WLMAppl (if workload management is in use) be located on the same dedicated machine. This arrangement keeps all management components physically adjacent and thus reduces the amount of intersystem DSOM traffic and provides plenty of processing capacity for Plex activation. Alternatively, use a system with low CPU utilization to ensure that the SMappl is unlikely to be CPU constrained. No performance measurements have been done to date looking at the CPU cost of the SMappl. The cost to a large extent depends on the size of the configuration, that is, the number of CICS systems and the number of resources within them. The most intense period of CPU activity is during a Plex activation.

- Management domains

You may well choose to divide your existing CICS systems into more than one grouping. For example, you may decide to run the test and production systems separately, each with its own CICS SM systems to manage them. Whatever the split is in your configuration, ensure that no CICS system is managed by more than one SMappl.

- Default routing

  Give some thought to what happens if workload management cannot make a routing decision, for example, if there is no active region or the required program or transaction is not enabled in any of the CICS systems. Workload management only makes its decision on the basis of available information. It cannot make resources available that are disabled for some reason or start CICS systems that are down. If a decision cannot be made by workload management, the CICS system that was specified before the invocation of workload management is used as default routing. By correctly managing the CICS systems and resources, default routing should not happen often, but you do need to be aware of it. It is advisable to have some policy in effect to cater for such a situation. You could, for example, resort to static routing. The problem of default routing also must be addressed with session-level load balancing.

- Which CICS systems or clients are to use workload management

  Decide which components of the configuration will use workload management so that you can determine where the workload management cache managers will be located. You also have to ensure that the workload management client is activated on these systems. We recommend that work be routed as close to its origin as possible, that is, route from an ECI version of the CICS Client for AIX rather than have the client attach to a region where you then make a routing decision.

- Workload management parameters

  Supply values for *Execution time on a reference system* for each program and transaction that is to be workload managed and for the *speed of this system relative to the reference system*. For the other workload management parameters, we strongly recommend that you work with the default values.

- CICS changes

  In those CICS systems that are managed by CICS SM, you may want to increase the number of application servers by 1 to allow for the processing work done by the agent transaction. The agent transaction is pseudo conversational and so does not use an application server all the time. However, if your configuration is highly tuned, you will find that there may be one less application server available than expected when the agent transaction is processing.

  We recommend that you run with a fixed number of application servers in a region; that is, set the minimum number of application servers to be the same as the maximum number. This setting is specified in the system model and system clone. By fixing the number of application servers, it will not be necessary for CICS to start an additional application server if a piece of work is routed to the region and the number of active servers has fallen below the maximum. No work has been done on the effect that starting additional application servers has on managed work and the consequent impact on future routing decisions.

- Performance considerations

  The design of the CICS SM Workload Manager is such that the overhead of a routing decision is made in the client component. This may physically be a

CICS Client for AIX or an AIX CICS region. The overhead that is observed in the client to a large extent depends on the situation.

For an ECI CICS Client for AIX, the overhead of using workload management depends on the complexity. The more complex the client, the lower the overhead. Measurements with a very simple ECI client have shown an increase in CPU consumption of approximately 20% and no overhead on the machine running the CICS system. It is important to recognize that the base cost, that is, the non-Workload-Manager case, was small because of the simple nature of the ECI program. With a more realistic ECI client program, you would see a lower overhead.

We observed an increase of approximately 12% in CPU consumption in a dynamic transaction routing measurement when the CICS for AIX region was used solely for routing work to one of three back-end CICS for AIX regions. Again there was no overhead in the routed-to systems.

The overhead of workload-managed DPL requests has not been measured. This overhead is similar in nature to the dynamic transaction routing case.

For session balancing support, there is only one call to workload management: when the client initiates a session with a CICS for AIX region. You may see a slightly extended initial access time, but after that there should be no impact.

The only CICS SM components running in the server system are the resource controller and the agent transaction. During periods of activation or definition, the resource controllers consume CPU. Once this activity has completed, negligible activity takes place in the resource controller. The agent transaction within a CICS region wakes up every 5 seconds and looks at the health of the CICS system. Both components have no measurable effect on server performance when there is no CICS region activation or definition work.

8.7 Summary

CICS SM provides a very effective tool with which to manage your CICS for AIX configuration. The major functions that CICS SM provides are SPOC, SSI, and workload management.

SPOC and SSI simplify the costs of administering your CICS for AIX configuration and thereby reduce the costs. The staff administering the CICS for AIX configuration no longer has to be aware of the location of the physical machines. It is possible to act on groups of CICS for AIX systems or just one single CICS for AIX system. It is not necessary for each of those machines to be accessed individually in order to define, start, or stop CICS for AIX regions or Encina SFSs.

With workload management it is possible to distribute the incoming work across the CICS for AIX regions capable of supporting that type of work. It is possible to improve the availability of applications to their end users by configuring your environment in a suitable manner. Managing system outage and applying maintenance can be simplified, while still providing a service to the users. These are important considerations, which are often overlooked in planning a new configuration. Minimizing the number and duration of outages is becoming increasingly important as commercial pressures increase on enterprises.

Functions such as the discovery of existing CICS for AIX systems, event monitoring, which enables you to monitor the health of regions or resources, and a GUI increase the value and usability of CICS SM. The discovery of existing CICS for AIX systems means that it is much easier to get started with CICS SM while building a CICS for AIX environment. You do not have to spend hours at the beginning building the necessary definitions of your CICS for AIX systems. Simply install CICS SM, direct it to discover which CICS for AIX resources currently exist, and then work from there.

While it is possible to start using CICS SM on your existing CICS for AIX configuration as is, to get the maximum return from the use of CICS SM, forward planning is required and possibly some reconfiguration. CICS SM can add significant value to the management of your CICS for AIX configuration and the service that it provides to users, be they employees or customers of your enterprise.

Chapter 9. Accessing CICS from the Web

The Internet is a federation of thousands of computer networks, with millions of attached computers that use a common set of technical protocols to create a global communications and processing medium. The Internet has an estimated population of 30 million users and a presence in more than 70 countries. This enormous community can have full-time, part-time, or dial-up connections to the network.

In recent years the Internet has grown in size and range at a greater rate than anyone could have predicted. In particular, more than one-half of the hosts now connected to the Internet run commercial applications. This is an area of potential and actual conflict with the initial aims of the Internet, which were to foster open communications between academic and research institutions. However, the continued growth in commercial use of the Internet is inevitable, so it will be helpful to understand how this evolution is taking place.

Hundreds of thousands of companies now make their business by relying on Internet users. They not only collect market information from the network but also conclude transactions with their customers on the other end of the network. The Internet has enabled them to find more business potential and enormously expand their business. More and more companies are expanding their applications from closed networks such as LANs to the Internet.

This explosion in the use of the Internet has largely been fueled by the availability of GUIs, thus making it much easier for non-computer-literate users to access the information that is available within the Internet.

The World Wide Web (WWW, W3, or simply the Web) is a hypertext, page-based, graphical information service. It provides access to multimedia and complex documents and databases. It is one of the most effective methods of information display and update because of its visual impact and such advanced features as:

- Ability to link to other “pages”
- Ability to run programs
- Ability to present forms, pictures, sounds, and movies
- Platform and operating system independence

Many enterprises recognize the commercial potential of the Internet and are seeking a means to establish a presence. But first they must develop the necessary applications in a timely manner. The CICS Internet Gateway for AIX can provide a means of accessing existing CICS applications. The use of the CICS Internet Gateway for AIX can significantly reduce the effort required to make applications available for use on the Internet. Another benefit of using the CICS Internet Gateway for AIX to access your CICS regions is that it is possible to use exisiting CICS skills to develop new CICS applications for the Internet without having to retrain staff.

9.1 Some Terminology

You have to understand a number of Internet and Web browser terms before we can discuss the CICS Internet Gateway for AIX in any level of detail.

URL A uniform resource locator (URL) is an addressing scheme. URLs are used to locate pages on the Web.

HTML The Hypertext Markup Language (HTML) defines the format and contents of documents sent from the Web server to a Web browser for formatting and display. HTML uses tags to specify formatting and document structure and identify hypertext links.

HTTP The Hypertext Transfer Protocol (HTTP) is the protocol that a Web browser uses to communicate with a Web server.

Forms Initially, the information flow was essentially one-way (from Web server to Web browser) on the Internet. A forms function has been added to HTML that allows you to specify input fields that enable Web browsers to send data to the server. It allows the Web browsers to access many different types of applications, such as OLTP applications.

CGI The common gateway interface (CGI) is a means of allowing a Web server to execute a program that you provide, rather than retrieving a file. For some applications, for example, displaying information from a databasee, you must do more than simply retrieve an HTML document from a disk. So the Web server has to call a program to generate the HTML to be displayed.

9.2 Services

The CICS Internet Gateway for AIX (see Figure 40 on page 135 and Figure 41 on page 136) makes it possible to access and run many existing and new CICS 3270 applications using standard Web browsers. Typically, no changes are needed in the CICS applications, and application code is not required for the gateway itself. The gateway can also be used to provide an alternative user interface for CICS applications within an enterprise as well as across the Internet.

The CICS Internet Gateway for AIX provides an interface between a Web server and a CICS application, allowing conversion of 3270 data streams into the HTML format used by the Web. The CICS application can then be accessed by any Web browser.

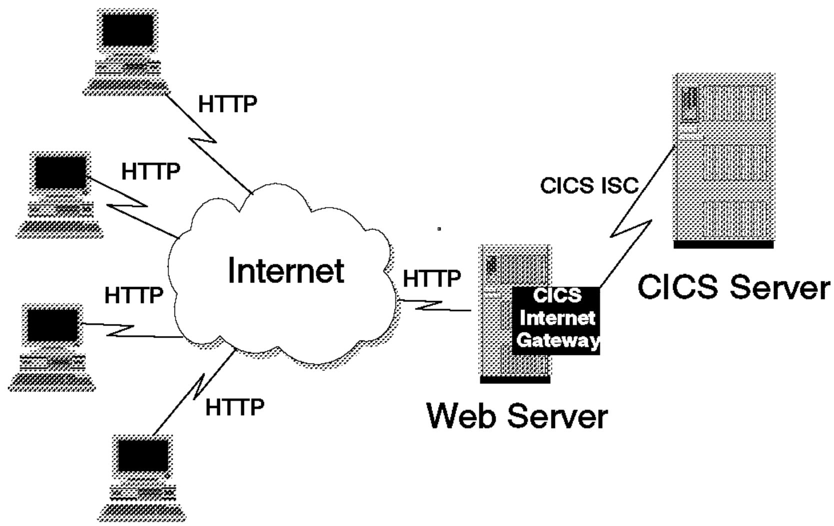

Figure 40. CICS Internet Gateway

The CICS Internet Gateway for AIX is a CGI script that takes HTML input from the Web browser, maps it to the 3270 data stream, and uses the CICS EPI to send it to a CICS server. The CICS Internet Gateway for AIX then intercepts the 3270 data stream returned by the CICS application and formats it for the Web browser.

Web browsers (running on an end user′s terminal) can directly access many Internet information services, but some complex data formats and commercial applications often need some help.

The CICS Internet Gateway for AIX enhances the functions of the Web server by exploiting the CGI. The CICS Internet Gateway for AIX dynamically builds Web pages containing the CICS data and transmits them to the Web browser.

The CICS Internet Gateway for AIX provides automatic translation between CICS 3270 data streams and HTML, using the CICS EPI, which is part of the CICS Client.

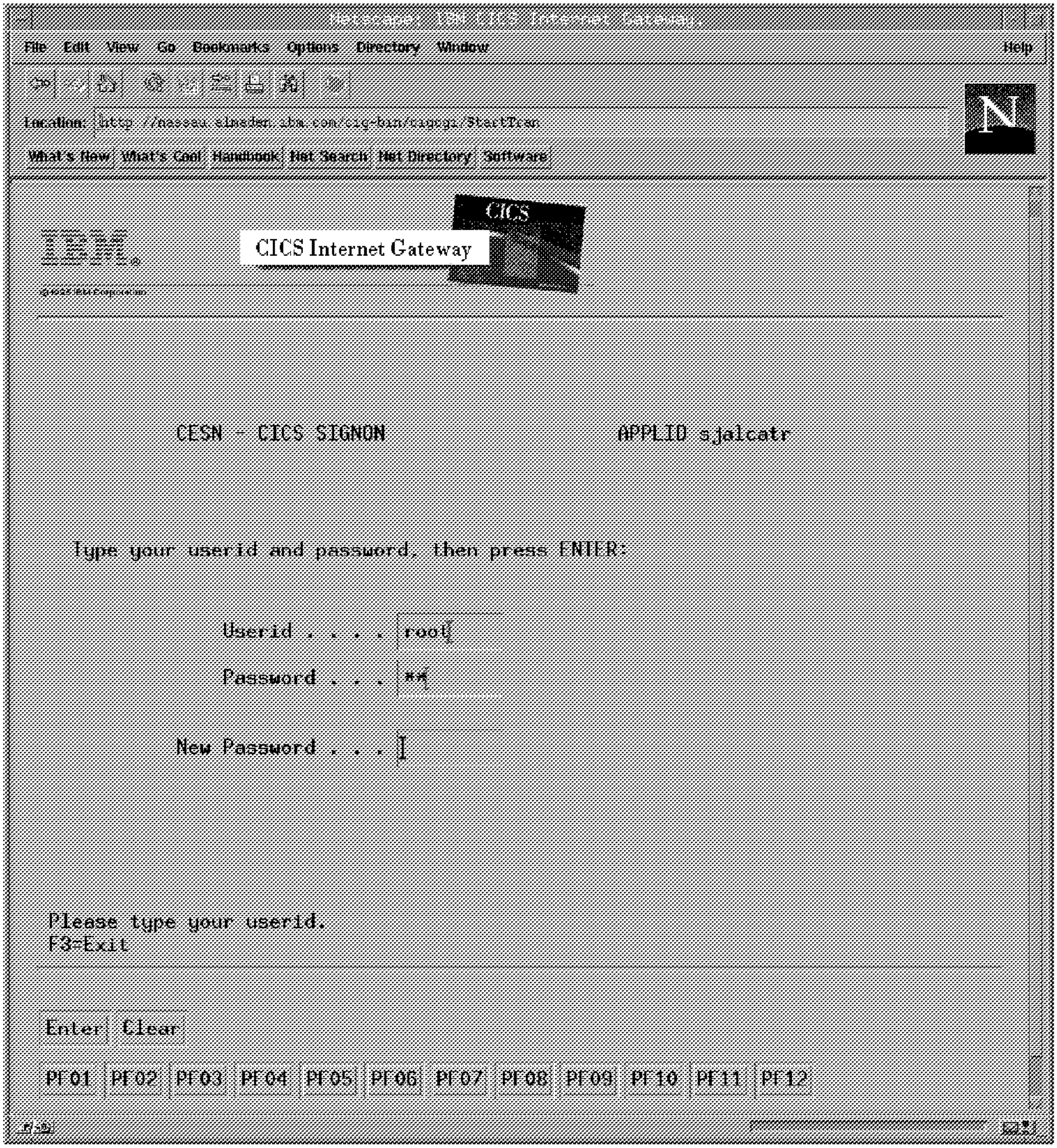

Figure 41. CICS Internet Gateway for AIX: Sample User Interface

9.3 OLTP and the Web

The processing that takes place on the Web differs in a number of ways from the processing in a typical OLTP environment. It is important to understand the differences between these two types of processing so that the applications you provide are suitable or have adequate recovery processing, for example.

The Web is designed to allow you to easily locate and *read* documents and other information that might be located on a system anywhere on the Internet. OLTP systems are designed to provide a highly reliable means to *read and update* centrally managed databases on which enterprises depend for their daily operation.

There are, however, some important similarities between Web servers and traditional transaction monitors:

- Both use standard data streams between the client and server—HTTP and HTML for Web servers or IBM 3270 and DEC VT series for OLTP, for example.
- Both schedule and process short repetitive pieces of work on behalf of many users accessing shared data.
- In both cases, the application runs on the server, and the client is dedicated to presentation.

As the Web is used to access traditional OLTP applications, the challenge is to combine the ease of use and flexibility of the Web with the robustness of the traditional OLTP environment.

9.3.1 Higher User Expectation

When you use the Web, you are typically searching for information. You might be looking for a specific piece of information, or just looking to see what is there. If you select a hypertext link, and it is not available, or the content of the link is not useful, you simply look elsewhere for something that better meets your needs.

When using an OLTP system, you are normally carrying out specific functions required to perform tasks that are part of your job. If a particular function does not work, you cannot successfully complete a task that is part of your job.

Clearly, the OLTP user has a much higher level of expectation about the consistency of system behavior and its reliability. Therefore when you design network and server systems, make sure that your systems are as reliable and consistent as possible.

9.3.2 User Interface

Web browsers provide a GUI that is easy to use and intuitive. You can display documents that are both large and small. If the document is larger than the current size of your Web browser window, you can easily scroll backward and forward through the document.

OLTP systems often use character-based, fixed-function display terminals. The user interface is typically optimized for efficient use by experienced users. Screen layouts are constrained by the number of rows and columns that the terminal can display. Where more data must be displayed than can fit on a screen, you may have to issue additional transactions to retrieve and display the additional data.

Although a user can access the existing CICS application with the original user interface, without any change through the CICS Internet Gateway for AIX, you should think about whether the original user interface is friendly or not when viewed by a Web user. You should also consider the impact of increased delays in providing transaction input such as the result of additional delays due to the Web browser or network. Accordingly, you may have to optimize the user interface, taking as much advantage as possible from the Web browsers.

9.3.3 Data Integrity

The Web was originally designed to enable you to retrieve and view documents available on the Web. In a read-only environment, you do not have to be concerned about recovery from failures when updating data. Consequently, Web servers do not provide facilities related to the integrity of data (HTML documents) that they access.

OLTP systems, in contrast, take responsibility for the physical and logical integrity of the data they manage, using facilities such as logging, transaction backout, and deadlock detection, either directly or in cooperation with a database management system.

OLTP systems can provide end-to-end recovery, ensuring that the user′s terminal has successfully received and displayed the transaction response before database updates are committed. Web-based applications typically end before the Web server attempts to send a response to the Web browser. They have no way of telling whether the response arrived successfully or not.

For 3270 sessions, CICS provides recovery facilities that enable you to ensure that no data sent to or received from a 3270 terminal is lost. CICS decides when to commit changes to its resources.

CICS clients using the ECI can manage their own data integrity by using the ECI_EXTENDED or ECI_NO_EXTEND parameters to control the scope of the LUW. See *CICS Family: Client/Server Programming*, SC33-1435-00, for more details.

The Web cannot provide the same level of data integrity between the Web server and Web browser .So with a Web application you have at least two connections to consider:

- The link between your Web browser and your Web server
- The link between your Web server and CICS server

9.3.3.1 Data Integrity between Web Browser and Web Server

If your Web browser and Web server are connected across the Internet, data transmission between them is not guaranteed. If your Web browser issues a request and receives a response indicating that the function completed successfully, you can be confident that the CICS transaction performed the function that you requested. If your Web browser does not receive a response, there is no way of knowing whether a CICS function that you requested was completed normally:

- It may not have run.
- It may have started and been completed normally, but the connection to the Web browser was lost before the results could be returned.
- It may have been abended.
- It may have been started and completed, but with the wrong result.

Although you will not normally be maintaining data resources on your Web browser, you can still design the CICS and Web server components of your application to ensure data integrity and consistency.

9.3.3.2 Data Integrity between the Web Server and CICS Server

The data integrity issues for the connection between the Web server and CICS server are no different from those for any other distributed CICS application. For example, you can use the ECI to manage an LUW that can span several calls to a CICS server program.

9.3.4 Processing State Information

In a transaction processing environment, a business function can consist of several related transactions that you perform in a particular logical sequence. The application therefore needs knowledge of the current state of processing within the logical business function that is being performed. It also must be able to distinguish between different users performing the same tasks at the same time.

When you use a Web browser, each time you select a new hypertext link, you might be accessing a different Web server. The Web server itself is stateless; it treats each request separately and independently and has no concept that a series of user interactions with the system can be related and make up a larger logical function. If you need to maintain information about the state of processing in an application using the Web, you must maintain information about the state of processing either with the Web browser or on the Web server, using programming interfaces such as the CGI.

OLTP systems understand state and provide facilities to help you keep track of state within your applications. CICS, for example, allows one transaction to specify which transaction is to process the next input transaction from a user. It also provides facilities such as the CICS COMMAREA and temporary storage queues that enable you to easily maintain state information and pass it from transaction to transaction.

9.3.4.1 Saving Processing State Information

You may want a mechanism that enables you to save processing state information that can be retrieved when new input is received for an ongoing Web conversation.

You first have to decide where you want to manage your information about the state of processing: on your Web server CGI script or the CICS server.

Web Server State Information: One method of minimizing the amount of information about the state of processing that has to flow across the network is to keep the information about the state of processing in storage on the Web server and allocate a unique identifier to that information. You can then send the unique identifier in the form, so that when the forms input is received, the Web server can use the unique identifier to retrieve the information about the state of processing for that conversation.

You may decide to do some or all of your state management on the Web server. One of the advantages of doing at least some of the state handling on your Web server is that it can perform a routing function. If you have multiple CICS servers, you can put code into your CGI script to analyze information about the state of processing returned by the Web browser and invoke the appropriate CICS server for the requested function. Thus your Web server can provide a routing function similar to that provided by a CICS TOR for conventional CICS transactions.

The drawback of performing state management on your Web server is that the operating system environment may not be as well suited to this function as the CICS environment. State management is not a new problem for the traditional OLTP environment. CICS provides a variety of facilities that can be used to manage information about the state of processing for ongoing transactions, most of which are designed to make it easier to write pseudo conversational applications.

CICS Server State Information: Using CICS temporary storage is a common method of storing information about the state of processing. Naming conventions can be used to allocate unique identifiers to each CICS temporary storage queue. One of the most common techniques used to allocate unique identifiers is to incorporate the CICS terminal identifier in the temporary storage queue name.

When a CICS terminal-oriented transaction ends, the next transaction to be run at that terminal can be specified either by the CICS program on its EXEC CICS RETURN statement, or in the CICS terminal definition for that terminal.

Managing information for the Web environment differs from managing information for the normal CICS environment in one important respect. In the traditional 3270 CICS environment, transactions are terminal-oriented. CICS associates information about the state of processing with a particular terminal, and many of the state management facilities that CICS provides are terminal-oriented. In the case of Web transactions that have been initiated through CICS DPL, no terminal is associated with the transaction, so these state management facilities cannot always be used. We therefore have to look at alternative methods of managing information about the state of processing for Web transactions (or for any other DPL or ECI application that does not have an associated terminal) that have save information about the state of processing across transactions.

9.3.5 High Availability

Few Web servers run mission-critical applications, at least not yet, and when a Web server is unavailable for any reason, it is unlikely to prevent a major organization from carrying on its function.

Many OLTP systems run mission-critical applications, however, and, when down, can prevent an organization from operating normally. Not surprisingly, OLTP systems such as CICS include high-availability features that enable them to recover from failures as rapidly as possible with minimal disruption to users.

9.4 Security Issues on the Internet

When you place information on the Web or the Internet, you must think about the security of that information. The Internet is not designed to protect confidential or sensitive information. Internet providers and users are not regulated. Anyone can use the Internet or become an Internet provider. Internet users have no control over the path across which their information flows. Since the Internet is a cooperative, shared participation network, there is nothing to prevent reading, copying, or changing information. There is no guarantee that information will ever reach its destination. Computers connected to the Internet can be accessed by anyone. Unless some form of security is used, both information being communicated and resources used for communication are unprotected from mischief or malice.

In general, OLTP systems are built on private networks where you can more easily protect the privacy of your data. However, as business use of the Internet grows and more and more private networks are linked to the Internet, the need for secure communications and networks grows as well.

Internet Security actually consists of two distinct services: network security and transaction security.

Network security: Network security refers to a corporation′s ability to protect its computers, memory, disk, printers, and other computing equipment from unauthorized use. Network security protects against attackers who try to access information or gain control over machines or resources within a private network.

The most common way of protecting private networks connected to the Internet from attacks is with firewalls, single points of connection between a private and public network that allow communications between them to be monitored and secured. Firewalls differ in their implementation and the degree of protection they offer. The most common types of firewalls are:

- **Screening filter**, which uses a router to connect the private network to the Internet. The screening filter monitors each IP packet flowing through it, controlling access on both sides. A screening filter cannot control access at the application layer, however.
- **Bastion**, a machine placed between the private network and the Internet that breaks the connection between the two. The bastion relays messages to or from authorized users and denies access to all others. Bastions can control access at the user or application layer, but they can be costly if many users are supported. If an attacker can impersonate an authorized user, the attacker can get into the private network.
- **Dual-homed gateway**, which combines a screening filter and a bastion into either a single machine or a series of machines. The gateway can be complex and make it hard to find attackers. Alternatively, screening filters can be used to protect bastions, or a combination of screening filters and bastions can be used to tailor protection to the subnet or resource being protected.

A more detailed discussion of Internet firewalls is beyond the scope of this book. For information see the following URL:

`http://www.raleigh.ibm.com/icf/icfprod.html`

Transaction security: Transaction Security refers to the ability of two entities on the Internet to conduct a transaction privately, and with authentication through digital signatures if required.

There are two aspects to transaction security:

- Authentication. For some users of the Web (in electronic commerce, for example) it is important that users authenticate themselves to Web servers, that Web servers authenticate themselves to users, or that both authenticate to each other. Whatever the form of authentication, it must not be easily compromised.
- Encryption. For applications in which Web clients and servers exchange sensitive information, such as userid and password pairs, credit card details,

or employee records, eavesdropping must be prevented through appropriate cryptographic techniques.

The authorization protocols supported by the Web are still evolving, and there is no widely accepted implementation. Two security packages have been around for a while: *Pretty Good Privacy* and *Kerberos*. Two relatively new transaction security standards are likely to become integral parts of a future integrated security solution for the Web: secure HTTP and secure sockets. See the following URLs:

```
http://www.netscape.com/newsref/std/SSL.html
http://www.eit.com/projects/s-http/index.html
http://www.ofs.org/www/dceweb/DCE-Web-Home-Page.html
http://hoohoo.ncsa.uiuc.edu/doce/PEMPGP.html
http://www.ibm.com/Security/glossary.html
```

The IBM Internet Connection Secured Network Gateway (SNG) is an effective tool to protect your network from unauthorized access. It is based on research at IBM's Yorktown Research Laboratory and experience running large networks for more than eight years. SNG supports all of the implementations above.

The IBM Internet Connection Secure Servers and the Secure WebExplorer browsers ensure the validity of the parties in a transaction and keep communications private.

9.5 Planning Considerations for the CICS Internet Gateway for AIX

A presence on the Internet is intended to give competive advantage to an enterprise, not compromise its security. Therefore you must carefully plan your approach to making applications available from the Internet and address a few configuration issues for implementing the CICS Internet Gateway for AIX.

9.5.1 Preparation

Before implementing the CICS Internet Gateway for AIX, become familiar with what the Web is and is not and understand what the differences between OLTP and the Web mean for your applications (see 9.3, "OLTP and the Web" on page 136).

9.5.2 Choosing Your Applications

Decide which applications and data are suitable for use on the Internet. The applications should be informative and helpful. It may be neccesary to redevelop applications, possibly focusing on the user interface. The typical user is likely to be far less computer literate than the employees of the your enterprise who were previously using the application

9.5.3 Accessing Sensitive Data

Determine whether sensitive data in the CICS system in which the Internet-accessible applications are running can be accessed. You may want to provide an isolated system whose sole purpose is to run the Internet-accessible applications.

9.5.4 Availability

Any Internet-accessible applications should be highly available so that they can be accessed by anyone at any time. A service that is unavailable does not create a good impression for your enterprise.

9.5.5 Operations

Ensure that the operation of any new CICS service that is created to support Internet-accessible applications is integrated into existing operations procedures.

9.5.6 Capacity

Regularly monitor the level of use of Web-accessible applications to ensure that there is sufficient processing and network capacity to cope with the volume of work. If you are using the application to collect marketing intelligence, for example, users may not be patient enough to wait for responses if the application is slow because of insufficient capacity of either the Web or CICS server. The objective of being connected to the Web has been lost for the enterprise.

9.5.7 Performance Considerations

The CICS Internet Gateway for AIX runs as a single process CICS EPI client (see Chapter 4, “Using CICS Clients” on page 49) to establish a terminal connection to the CICS for AIX region. The number of concurrent active connections could be constrained by:

- Web server limits on the number of concurrent active users or CGI scripts
- CICS Internet Gateway for AIX limits. There is a limit of 240 concurrently active users awaiting a response.
- CICS Client limits on the number of active CICS terminals per process. An Internet user is the same as a terminal in this case.
- Operating system limits on the number of threads per process. Each user uses up to two threads.

9.5.7.1 Restrictions and Limitations

Be aware that the CICS Internet Gateway for AIX does not support ATI requests for terminals. It uses dynamic terminals and so cannot support this function.

9.6 Summary

Establishing a presence on the Internet can be a difficult and complex process. This is a new area of technology where skills are still scarce. One critical success factor in establishing a presence is the creation of the necessary applications that Internet users will access. The CICS Internet Gateway for AIX enables an enterprise to make existing CICS for AIX applications acessible to Internet users, typically without modification. You may want to modify those applications to restrict the options or functions. That is a business decision, however. Because existing CICS skills can be used to develop new applications, application development is rapid and inexpensive.

OLTP and the Web are different forms of processing. There are similarities, but there are also significant differences in the type of conversation and in recovery and security considerations. It is important that you understand the differences and act accordingly.

Chapter 10. Application Design Considerations

OLTP systems can provide accurate, up-to-date information within seconds, from terminals that can give direct access to data held either as files or databases. Developing such a system would be a major undertaking, particularly if you had to write all of your own control programs for handling terminals, files, and databases and provide your own transaction processing mechanisms. However, Transaction Server for AIX supplies transaction processing and resource management functions that enable you to concentrate on developing application programs to meet your enterprise′s business needs.

10.1 Functions

CICS controls OLTP application programs in a DTP environment. CICS handles interactions between terminal users and your application programs. Your programs gain access to the CICS facilities with straightforward, high-level commands. CICS provides:

- Communication functions to terminals and systems required by application programs
- Control of concurrently running programs serving online users
- Facilities for accessing databases and files
- The ability to communicate with other CICS family members using LU 6.2 and TCP/IP
- Interactive facilities to configure your system
- Recovery processing and data protection, should a problem occur

Online application programs have certain common features and needs. Typically, they:

- Serve many online users, apparently simultaneously
- Require access to the same files and databases
- Try to give each user a timely response to each interaction
- Involve communications to terminals

CICS works alongside the operating system, providing the services required by application programs.

10.2 OLTP Application Development Life Cycle

Application program design is an iterative process. Decisions about the user interface affect transaction definitions, which in turn cause a slight change in specifications, and the whole cycle begins again. A simple development life cycle can include steps to:

1. Define the problem: Broadly specify the function required. This will come from the user departments within your organization.
2. Design the transactions: You have to define transactions and LUWs to perform the defined functions. Consider these points:
    - Requirements imposed by the environment, including terminal type and size, data storage format, and security requirements
    - Machine usage, response times, and transaction availability

- Type, size, and quantity of records to be processed
- The IBM 3270 Information Display System that CICS emulates. This is how your applications communicate with the application user.
- Efficiency and usability of the application. These are major goals in any application.
- Screen layout of your user interface, consistency between screens, number of expected keystrokes, and number of confirmation messages issued to reassure users
- Difference in constraints between online processing and batch applications
- Exceptional conditions that come with new considerations for an online environment
- Level of programming specifications and the availability of up-to-date system information

3. Write the application programs. Considerations include:
   - Which programming language
   - Defining screens with BMS
   - How your application programs use the BMS screens
   - Saving data and communicating between transactions
4. Translate and compile the application program.
5. Test the application program.

10.3 Developing Client-Server Applications

In CICS on Open Systems, the ECI and EPI enable non-CICS applications in a client system to use the facilities of CICS in a connected server system.

The ECI and EPI interfaces provide CICS client support for a non-CICS application. Thus non-CICS applications do not have to be on the same machine as the CICS for AIX region. Non-CICS programs using the ECI and EPI can be distributed across your configuration in the same way that clients like cicsterm can be. These interfaces enable you to combine the most up-to-date user interfaces with access to a CICS server. The presentation logic can run on the client machine, while the business logic and data management are dealt with on the server. One interface may be able to act as a front end to several CICS systems. However, the user is unaware that multiple CICS systems are being accessed. There is no need for the users to perform the switch from one system to another; the ECI or EPI application will do it for them.

10.3.1 The External Call Interface

With the ECI, you can write applications that:

- Call a CICS program in a CICS server from a non-CICS program
- Connect to several servers at the same time
- Have several outstanding program calls at the same time
- Access CICS programs, files, transient data queues, temporary storage, and transactions
- Exchange data between the client and the server
- Can be workload managed using CICS SM. See Chapter 8, "Using CICS System Manager for AIX" on page 111.

The ECI application programs make synchronous or asynchronous calls. Synchronous calls return control when the called program completes; the information returned is immediately available. Asynchronous calls return control without reference to the completion of the called program, and the application is notified when the information becomes available. Calls may be extended, that is a single LUW may cover more than one successive call, although only one call can be active for each LUW at a time. The application can manage multiple LUWs concurrently if it uses asynchronous calls.

The called program can:

- Update resources on its own system
- Use DPL to call CICS programs on other systems.
- Access resources on other CICS systems with function shipping or DTP.

The ECI consists of three types of calls:

- Program link calls that cause a CICS program to be executed on a CICS server.
- Status information calls that retrieve status information about the application and its connection to the CICS server.
- Reply solicitation calls that retrieve information after asynchronous program link or asynchronous status information calls.

Also available with the ECI is the ability to retrieve information about available servers to which the calls are directed.

10.3.2 The External Presentation Interface

With the EPI, you can write applications that:

- Allow a non-CICS application program to be viewed as a 3270 terminal by a CICS server system to which it is connected
- Connect to several servers at the same time
- Have several outstanding program calls at the same time
- Schedule transactions, where the application acts as the principal facility
- Balance sessions across the available CICS systems. See Chapter 8, "Using CICS System Manager for AIX" on page 111.

In CICS servers that support access through the EPI, other CICS transactions running in the server can use the START command to schedule transactions that will use the non-CICS application as their principal facility. When a transaction is initiated, 3270 data streams and events are passed between the CICS server and the application. The application can present the contents of the terminal I/O to its user in any manner appropriate to the application's operating environment.

Transactions may be routed to other CICS systems by standard transaction routing. Resources on other CICS systems can be accessed with function shipping.

The EPI consists of functions, data structures, and events. The EPI functions define application programming calls, such as installing new terminals to be controlled by the process, sending data from a terminal, and terminating the process. The EPI data structures define EPI data, such as reason codes, details of events, terminal details, and sense codes. The EPI events are used to respond to events that occur against a terminal, such as when a transaction sends data and is expecting a reply.

10.3.3 Writing ECI and EPI Application Programs

ECI and EPI application programs that run on CICS on Open Systems clients can be written in COBOL, C, or PL/I. Programs that do not make operating-specific calls are portable between CICS on Open Systems clients and other CICS client products. Application programs can use the facilities of both the ECI and the EPI.

To use the ECI and EPI, programs must be linked to the appropriate ECI and EPI libraries. Sample programs are supplied with auxiliary files that help with linking the programs.

For more information about the ECI and EPI interfaces see *CICS Family: Client/Server Programming*, SC33-1435-00.

10.4 How CICS Runs Your Transactions

The major components of the CICS for AIX run-time system are:

- The CICS client processes
- The transaction scheduler
- A pool of application servers
- CICS queue and file manager
- A PPC Gateway Server

See 3.2.1, "Components" on page 28 for more details.

There are two steps when CICS executes your transaction:

1. Requesting a transaction to be run
2. Transaction execution

When you enter a CICS transaction identifier through a CICS client, it sends an RPC to the transaction scheduler. The transaction scheduler selects an application server to run the transaction and requests it to do so with an RPC.

When you link-edit a CICS program to be run under CICS (for example, using the CICS-provided command, cicstcl, to translate, compile, and link), the object you create is not directly executable by the operating system because it contains some unresolved symbols and expects to be run by a CICS application server. The CICS application server provides a complete environment for running the loadable objects produced by cicstcl. To run your transaction, CICS looks up the name and location of its first CICS program and uses the operating system dynamic loading facility to load that program into the application server. The unresolved symbols in the program are then resolved by symbols provided by the server, and the program begins execution. The resolution of symbols applies irrespective of the language in which the program is written. Programs produced by the C compiler (or by the IBM COBOL or IBM PL/I compilers) are built as standard XCOFF objects and are thus shared automatically among multiple application servers on the same machine.

10.5 CICS Application Programming Interface

The CICS API is covered in some level of detail in 3.1.2, "CICS API" on page 24. See that section for details.

10.6 CICS Application Development Tools

CICS provides tools to develop and debug transactions. The tools cover a variety of areas from helping to design user screens to helping determine code problems. We discuss each of the tools:

10.6.1 Presentation Interface Development

You use the CICS BMS processor to translate BMS source files, which contain the definitions of map sets, to produce a symbolic map and a physical map. The symbolic map is a programming source language data structure (a COBOL, C, or PL/I structure) used by the compiler to resolve source language references to fields in the map. The physical map contains the information necessary to display the map on a physical terminal and instructions for embedding control characters within a data stream.

CICS provides the screen design aid (SDA), which is an interactive tool that allows you to easily design and modify BMS maps by using a Motif interface.

10.6.2 Application Program Translation

Application programs that include CICS API commands are processed by the command language translator (cicstran) that translates the CICS API commands into statements in the language used. This translator accepts as input a source program written in COBOL or C where CICS API commands are coded and produces as output an equivalent source program where each command is translated into statements in the language of the source program. You can then compile and link-edit your programs using the COBOL or C compilers.

Alternatively, you can request that your source program is translated, compiled, and link-edited in one step using the shell script cicstcl. The advantage of this alternative is that CICS uses the correct compilers and sets up the options required by CICS for translation.

The IBM PL/I compiler has an integrated CICS processor. Thus it is unnecessary (and indeed not possible) to run a separate CICS translator step for CICS programs written in IBM PL/I. You should therefore only use either cicstcl or the IBM PL/I invoked directly and not attempt to use cicstran.

10.6.3 Application Program Debugging

CICS provides a transaction called the Execution Diagnostic Facility (CEDF) that enables you to debug an application program that has been preprocessed with the -e option (on cicstran or cicstcl) without modifying the program. The facility displays the state of the application program at the interception point and allows you to interact with the debugging tool before returning control to the application code.

You can use the Lexington Debugger with C, IBM COBOL, and IBM PL/I programs.

The ANIMATOR debugging tool enables you to test a Micro Focus COBOL application program online without modifying the program. This tool intercepts execution of the application program at various points before displaying information about the program. Any screens sent by the application program are displayed by the tools, so that you can converse with the application program during testing just as you would on the production system.

The IBM Application Debugging Program provides the ability to debug IBM COBOL, C, or PL/I programs.

There are a number of new facilities to go with the debugging tool:

- A CICS-supplied transaction, CDCN, that turns the debugging tool on and off
- A regionwide attribute, AllowDebugging with settings yes and no, to control whether the debugging tool can be used within the region.
- Two transaction definitions (TD) entries:
  - DFHCDCN0 (for the CICS-supplied program)
  - DFHCDCN (for the CICS-supplied mapset)

10.6.4 Using Transactions to Call Your Program

In CICS, the term *transaction* is used to describe a fundamental unit of work, consisting of one or more application programs, that is initiated by a single request. A transaction is identified by a transaction identifier (tranid), which is used to call your application program in the CICS run-time environment.

Given that both the transaction and program are defined to CICS, you can use the cicsterm or cicsteld commands to run the transaction and therefore run your application program. The transaction is executed in the run-time environment under the control of CICS, which provides the services requested by each API command and dynamically changes the control parameters.

10.7 Designing Efficient Applications

As in all design, there is a balance to be struck between the possibly conflicting requirements of performance, efficiency, ease of maintenance, productivity, readability of code, usability, standards, budgets, and schedules. The standards and guidelines of your organization, and the prioritization of design goals, will determine how much of this section you can apply in the applications you are developing. It is assumed here that you have already completed your system design to a logical level and are now trying to divide the application processing into CICS transactions and programs, its input and output into BMS maps, and its data into CICS data areas, files, temporary storage queues, and transient data queues.

10.7.1 CICS Transaction Design

A primary consideration for online transactions is whether to design conversational or pseudo conversational transactions.

10.7.1.1 Pseudo Conversational and Conversational Transactions

Pseudo conversational programming is based on the fact that, in a conversational transaction, the length of time spent in processing each of a user′s responses is extremely short when compared to the amount of time waiting for the input. A conversational transaction is one that involves more than one input from the terminal, so that the transaction and the user enter into a kind of conversation. A nonconversational transaction has only one input (the one that causes the transaction to be invoked). It processes that input, responds to the terminal, and terminates.

In a conversational transaction, processor utilization times, even allowing for accessing files, are considerably shorter than terminal transmission times, which are considerably shorter than user response times. This is especially true if users have to think about the entry or enter many characters of input. Consequently, conversational transactions utilize storage and other resources for much longer than nonconversational transactions.

A pseudo conversational transaction sequence contains a series of nonconversational transactions that look to the user like a single conversational transaction involving several screens of input. Each transaction in the sequence handles one input, sends back the response, and terminates.

Before a pseudo conversational transaction terminates, it can pass data forward to be used by the next transaction initiated from the same terminal, whenever that transaction arrives. A pseudo conversational transaction can specify what the next transaction is to be, and it does this by setting the tranid of the transaction which is to handle the next input. However, be aware that if another transaction is started for that device, it may interrupt the pseudo conversational chain you have designed.

No transaction exists for the terminal from the time a response is written until the user sends the next input and CICS starts the next transaction to respond to it. Information that would normally be stored in the program between inputs is passed from one transaction in the sequence to the next using the COMMAREA or one of the other facilities that CICS provides for this purpose.

10.7.1.2 Correspondence of CICS Client Processes and Servers

In a typical CICS system there is likely to be a small pool of application servers servicing requests from a much larger pool of CICS cicsterm and cicsteld processes.

It is important to understand that a conversational transaction requires exclusive use of an application server, thereby reducing the total number of servers available to process other requests. CICS will create additional servers as dictated by system demand up to a maximum number that you can configure, but once you reach this limit no more will be created. Therefore if you design all of your applications to be conversational, there is a considerable risk that CICS will be unable to service user requests to run transactions.

10.7.1.3 Choosing between Pseudo Conversational and Conversational Design

There are four major issues to consider in choosing between pseudo conversational and conversational programming. The first of these is the effect of the transaction on contention resources, such as storage and processor usage. Storage is required for control blocks, data areas, and programs that make up a transaction, and the processor is required to start, process, and terminate tasks. Conversational programs have a very high impact on storage, because they last so long, relative to the sum of the transactions that would comprise an equivalent pseudo conversational sequence. However, there is less processor overhead, because only one transaction is initiated instead of one for every input.

The second issue is the effect on the use of exclusive resources such as records in recoverable files, recoverable transient data queues, and enqueue items. Again, a conversational transaction holds on to these resources for a very much longer time than the corresponding sequence of nonconversational transactions. From this point of view, pseudo conversational transactions are better for quick responses, but recovery and integrity implications may force you to use conversational transactions.

CICS ensures that changes to recoverable resources (such as files, external database, transient data, and temporary storage) made by an LUW will be made completely or not at all. An LUW is equivalent to a transaction, unless that transaction issues EXEC CICS SYNCPOINT commands, in which case, the EXEC CICS SYNCPOINT marks the end of one LUW and the start of the next. When a transaction makes a change to such a resource, CICS makes that resource unavailable to any other transaction that wants to change it until the original transaction has completed. In the case of a conversational transaction, the resources in question may be unavailable to other terminals for relatively long periods. For example, if one user tries to update a particular record in a recoverable file, and another user tries to do so before the first one finishes, the second user′s transaction is suspended. This has advantages and disadvantages. You would not want the second user to begin updating the record while the first user is changing it, because one of them will be working from what is about to become an obsolete version of the record, and these changes will erase the other user′s changes. However, you also do not want the second user to experience the long, unexplained wait that occurs when that transaction attempts to read for update the record that is being changed.

If you use pseudo conversational transactions, the resources are only very briefly unavailable (that is, during the short component transactions). However, unless all recoverable resources can be updated in just one of these transactions, recovery is impossible because LUWs cannot extend across transactions. So, if you cannot isolate updates to recoverable resources in this way, you must use conversational transactions.

The above example poses a further problem for pseudo conversational transactions. Although you could confine all updating to the final transaction of the sequence, there is nothing to prevent a second user from beginning an update transaction against the same record while the first user is still entering changes. Therefore you need additional application logic to ensure integrity. You can use some form of enqueuing, or you can have the transaction compare the original version of the record with the current version before actually applying the update.

The third issue concerns the relative performance profiles of the two transaction types. Although it is better for overall throughput to design your transactions to be pseudo conversational, conversational transactions use less processing power. The reasons are:

- It may be necessary for an application server to be created to run the transaction. This involves a large amount of processing because CICS must create an environment for the transaction that is persistent across failures. You should try to avoid application server creation by setting the value of the RD stanza attribute Minservers to a value appropriate for your expected workload.
- When each pseudo conversational transaction starts, the server has to be initialized for it. This initialization may include having to load the CICS program for it if it is not already loaded in shared memory.

The fourth issue is the delicate balance of transaction throughput and response times. In a heavily loaded system, once a conversational transaction is running, it will give more predictable response times than a pseudo conversational one. When the user presses an attention identifier (AID) key in the latter case, no application server may be available to process it, and the limit on the number of servers may have been reached. Conversely there should be better transaction throughput in the pseudo conversational case.

There are factors other than performance overhead to consider when choosing between pseudo conversational and conversational design for CICS applications, as described below. The method you choose can affect how you write the application programs. You may need extra CICS requests for pseudo conversations, particularly if you are updating recoverable files. Once you have done this, however, operational control (performance monitoring, capacity planning, recovery, system shutdown, and distributing system messages) may be much easier.

10.7.1.4 CICS Program Design

Having decided whether your transaction is best written to be conversational or pseudo conversational, your next task is to design it. Most organizations mandate a modular approach of some sort to this design task, and a primary consideration here is your choice of CICS on Open Systems, operating system, and programming language facilities.

10.7.2 Data Operations

The CICS file services API is designed to allow access to data files whose size can range from small to large, in a variety of file organizations and access methods, and which are usually of a more permanent nature.

The transient data and temporary storage service API commands allow access to sequential and random access queues.

The journal services API allows you to write journal records for audit trails or other purposes for offline processing.

In addition, CICS allows access to relational databases that provide a programmable interface through EXEC SQL commands in one of the supported CICS programming languages.

CICS user files (files defined within the File Definitions (FD) stanza) can be managed on either a DB2 Version 2 (CICS for AIX only) database or an Encina SFS. Either file manager can permit data to be shared between two or more regions and between regions and non-CICS applications such as batch processing programs.

When designing systems to exploit these facilities, you will have to consider the networking costs of transferring data between the file manager and its client, and the benefits of distributing the processing and disk access load between several machines.

CICS attempts to minimize network traffic by avoiding unnecessary interactions with the file manager. For example, at the end of a transaction CICS will retain an SFS open file descriptor for possible reuse where a region uses SFS for file access services. Such open file descriptors are released when the file is closed, either by an explicit EXEC CICS SET FILE CLOSED request or when CICS is shut down.

10.8 Summary

In developing applications for a CICS environment it is important that you understand the facilities that are available and the way in which CICS processes work so that you can work with the CICS architecture and not against it. In this chapter we discuss the key issues that you have to consider.

There are multiple ways in which to code applications. The style most suitable for your applications will depend on what those applications will do. CICS provides tools to assist in the development and debugging of applications. Those applications can be written in multiple languages.

The availability of the CICS ECI and EPI programming interfaces greatly improves your ability to extend client-server systems. You are no longer bound by the constraints of a 3270 data stream and BMS maps. GUIs as a front end to a CICS client are now possible, providing a very modern user interface.

Part 3. The Implementation

Having identified the problem, that is, how to develop and manage OLTP systems, and the solution, IBM Transaction Server for AIX, we now look at a sample environment to discover how IBM Transaction Server for AIX can help facilitate complex OLTP.

Chapter 11. TDJ Enterprise International Limited

The Disc Jockey (TDJ) Enterprise International Limited is a middle-sized company by international standards. Its annual revenue was 2.64 billion U.S. dollars in 1995, and its net profit was 271 million U.S. dollars. It is a holding company controlling three groups of companies which span the globe. They are:

- Lion Financial Group

  The companies comprising this group are Lion Insurance Limited and Lion Credit Union with their respective subsidiary companies. Their main lines of business are car and travel insurance as well as personal loans for holiday expenses or leisure.

- Kangaroo Leisure Group

  The companies comprising this group are Kangaroo Travel, Kangaroo Airlines, Kangaroo Hotel chain and their respective subsidiary companies. They are in the travel and leisure industries. They offer a wide spectrum of holiday packages ranging from the most luxurious to most the economical. Each client's needs are specially catered for by the enthusiastic well-trained staff of Kangaroo Leisure Group.

- Dragon Industries Group

  The companies comprising this group are Dragon Manufacturing Industries Limited and Dragon Music Company Limited with their respective subsidiary companies. The products manufactured are travel goods and music compact discs.

TDJ grows by acquisition as well as the establishment of new companies. The diversity of the industries that these companies serve is reflected in the IT equipment that they deploy to support their daily business needs. Before 1994, the bulk of the OLTP processing for the whole enterprise was done on three mainframe class systems running CICS/ESA regions with DB2 databases. In 1994 TDJ decided to move toward an open distributed OLTP environment across its whole enterprise network, integrating all of the existing IT equipment of the subsidiary companies, making itself ready to take on new IT technologies as they become available. Such a strategy meets the challenge imposed by the volatility and dynamics of the business environment in which TDJ competes.

As TDJ has been using CICS as its transaction monitor for many years, its complete satisfaction with CICS and a pool of in-house CICS skilled personnel influenced its decision to use the CICS family of products as the basis on which to build its new OLTP complex. CICS for AIX was finally chosen to be the core transaction monitor. In this chapter we want to share with you their experience in the implementation of their OLTP complex. We explore different parts of this OLTP complex, focusing in to show you the detail. You will see how features such as the high level of CICS family interconnection coupled with data integrity have helped TDJ implement its strategy.

11.1 The Big Picture

Figure 42 on page 159 shows you an overall view of TDJ′s OLTP complex. This is a complex, highly interconnected environment. It was not built by a single big project. Instead, it has grown organically as the business needs arose. At each stage of the growth existing investments were preserved and protected while at the same time pursuing a highly distributed open OLTP strategy. The core of the OLTP complex is made up of 16 CICS for AIX regions that provide transaction processing capabilities to serve the different business units within the enterprise. A variety of DB2 databases largely form the permanent storage for the enterprise data. The PCs and workstations of the end users form local LANs that are in turn connected to various servers to build a hierarchy of networks, which result in the enterprise network.

The three mainframe systems remain active and serve a number of end users through the 3270 terminals located at the corporate headquarters. They also serve as the archive servers for old data from other computers within the enterprise. Corporate legacy data and applications are now accessible by any authorized user across the whole of the enterprise network.

With the recent explosive growth in usage of the Internet and Web servers, Kangaroo Airlines established a Web server that is capable of obtaining flight booking information from the CICS transaction that runs under CICS for AIX. Thus the potential traveler can make his or her own inquiry about the price of an air ticket, flight schedules, or seat availability on a particular flight, all from a PC or workstation at any time of the day.

Within Kangaroo Leisure Group current data is stored in various databases that reside on a number of different platforms. The main databases are for airline reservation, hotel reservation, and travel booking. Access to these databases by a front-line staff is extremely important because the group advertises extensively that it provides friendly ONE STOP shopping for your next dream holiday!

The Lion Financial Group requires continuous availability for its car, travel, and house insurance systems so that its branch offices and agents around the globe always have access. The core machine is a RISC System/6000 SP (9076) with 16 nodes running seven CICS for AIX regions that access a parallel database. The group has started to set up a Web server to enable existing and potential clients to browse the new product offerings at their own pace. The ability to submit loan applications through the Internet is still under trial.

Dragon Industries Group has a number of manufacturing plants located in several countries. These plants have their own local computers that communicate with the Group′s central machine so that details such as current inventory, pending orders, and shipment details are readily available. Thus it is possible to track the current production status and identify any bottlenecks. The group has a "just-in-time" manufacturing policy and is actively pursuing ISO 9000 accreditation. Therefore it is vital to have the relevant procedures up to date and available online to all authorized staff.

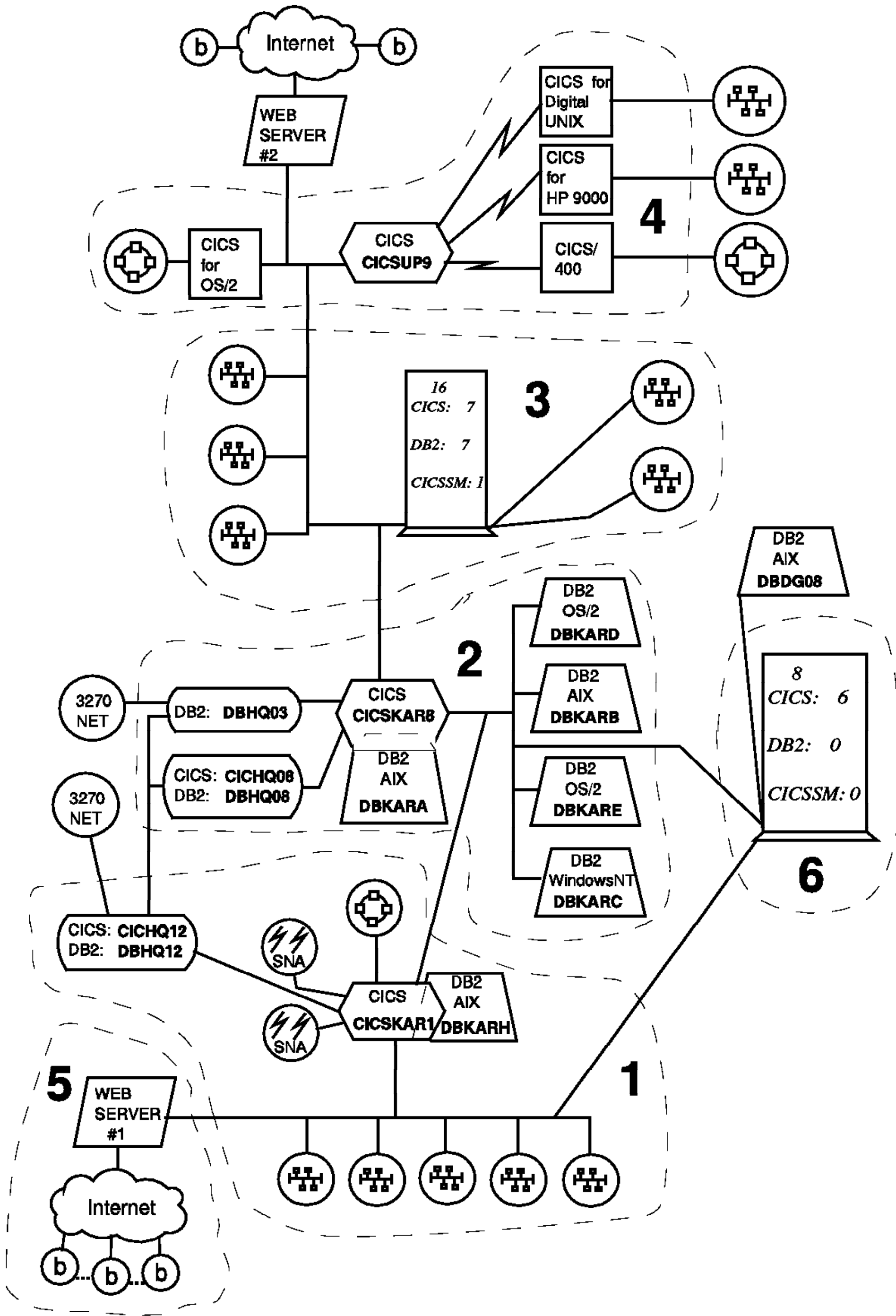

Figure 42. The OLTP Complex of TDJ Enterprise International Limited

The complex is divided into six areas. Each of these areas has been chosen to highlight how the IBM Software Server and CICS for AIX in particular have helped TDJ pursue its strategy of developing an open distributed OLTP environment.

11.2 User Access

Zooming in to sector 1 of TDJ′s OLTP complex (see Figure 43 on page 161), you can see the network connections that link the end users′ desktop PCs and workstations to CICS for AIX region CICKAR1. This region serves the business needs within the Kangaroo Leisure Group.

A significant number of CICS users have access to this region through the cicsterm or cicsteld facilities provided by the CICS Client for AIX. These cicsterm and cicsteld processes run in separate RISC System/6000 machines other than the server machine where the CICS for AIX region resides. Such an approach prevents the overloading of the server machine with these processes and enables CICKAR1 to deliver good transaction response times and throughput.

All end users have to authenticate to DCE before signing on to CICS and use the CESN transaction before they can initiate other CICS transactions. This authentication helps to enforce a more secure environment.

A small travel agency affiliated with Kangaroo Travel uses a SunSPARC Server system with the CICS Client for Sun Systems product to access CICS for AIX region CICKAR1. Thus the agency has the same level of information that is available to Kangaroo Travel staff and at the same time can provide good service to their clients and remain competitive.

A variety of PCs on local token-ring LANs connect to the RISC System/6000 system running CICS for AIX region CICK AR1. The connections are made through a gateway to the SNA adapter of the RISC System/6000 system or directly to the token-ring adapter of the RISC System/6000 system. With the appropriate version of CICS Client installed on the respective platforms, the end users of these PCs access the CICKAR1 region and can execute transactions. The appropriate LD has been created in CICKAR1 to support the communication from these CICS clients.

CICS for AIX region CICKAR1 is connected to CICS/ESA region CICHQ12 through CICS ISC so that a CICS transaction running in CICKAR1 can access the legacy data stored centrally in the mainframe computer.

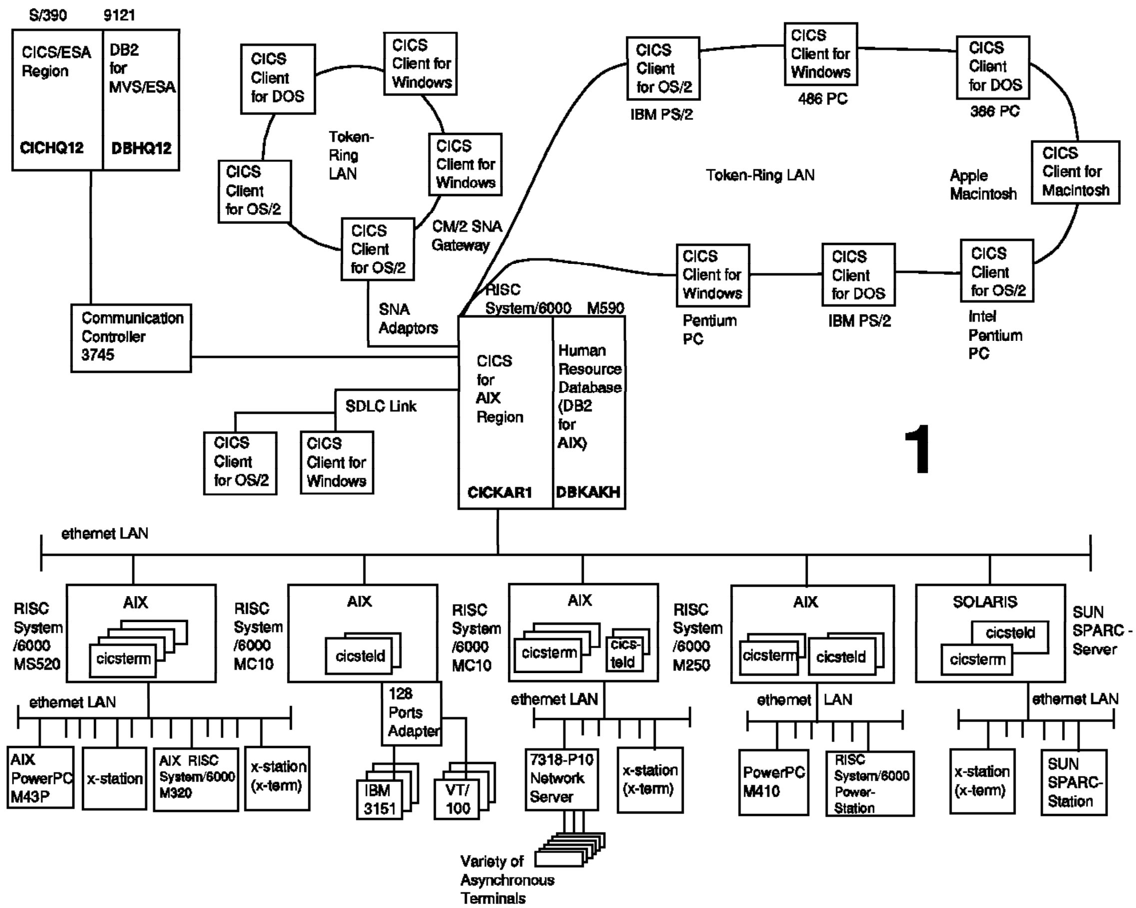

Figure 43. Accessing CICS Applications through CICS Clients

11.3 Databases

TDJ′s OLTP complex makes extensive use of DB2 databases. Figure 44 on page 162 shows the databases that are accessible from CICS for AIX region CICKAR8.

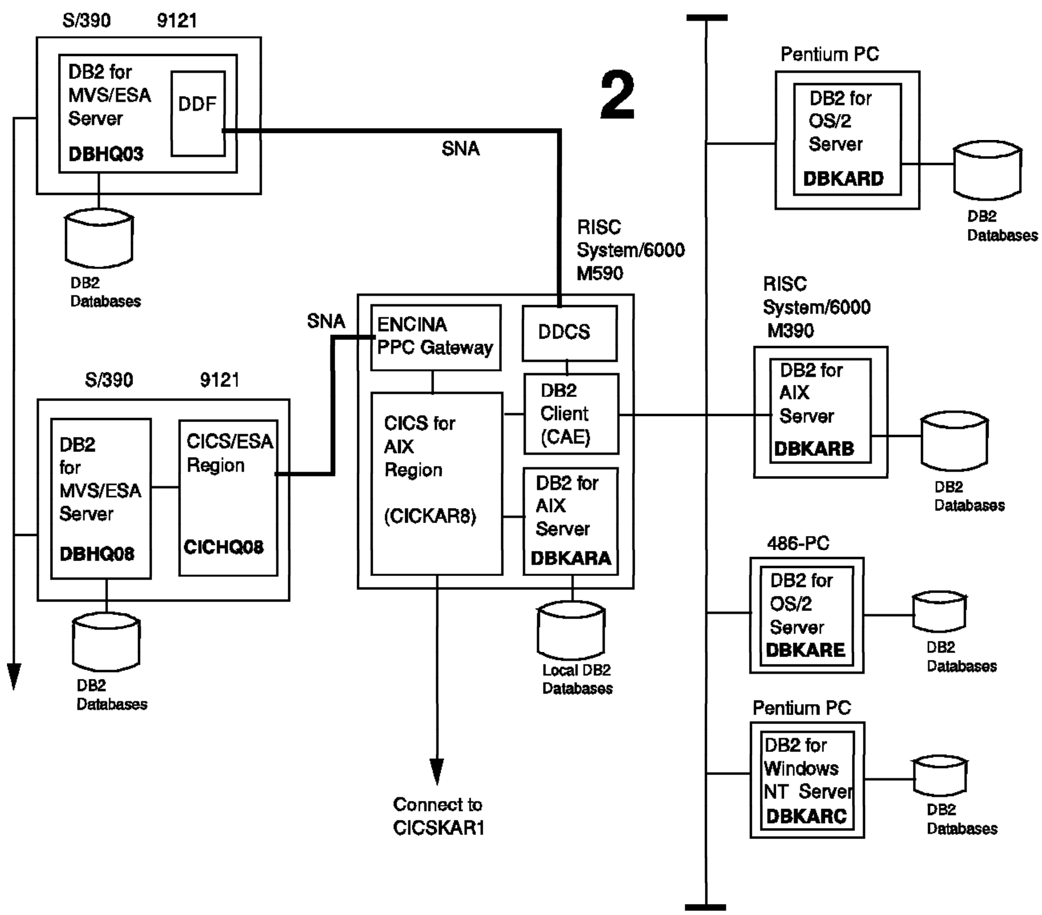

Figure 44. Access to Various DB2 Databases from CICS

CICS for AIX region CICKAR8 serves the Kangaroo Hotels Corporation within the Kangaroo Leisure Group. The main CICS application suites that run in this region are the Accommodation Booking System and the Guest Invoice System. The databases for these two applications reside in a separate RISC System/6000 system running an instance of DB2 for AIX server called DBKARB. This provides a dedicated database machine, which gives the required throughput for the busy applications that these databases serve.

There are some local DB2 for AIX databases that CICKAR8 can access. The use of these local databases is very light, so their presence in the same machine as CICKAR8 does not cause any noticeable impact on the performance of CICKAR8.

The Kangaroo Hotels Corporation is committed to providing a very high quality service to its hotel guests and patrons. It does not want to cause any frustration and anger among them due to data integrity problems with its IT systems. So, all the DB2 databases are XA enabled with two-phase commit. The single-phase commit optimization is not a viable option as updates are performed on other systems through the Encina PPC Gateway.

CICS for AIX region CICKAR8 still uses a local Encina SFS as its file manager, not DB2 for AIX. The fact that nonrecoverable CICS resources are supported as recoverable resources on DB2 would require the rewriting of several existing CICS applications. This requires significant programming effort; hence it was decided not to use DB2 for AIX as the CICS queue or file manager. The decision may be reviewed in the future.

Other DB2 databases reside on OS/2 and Windows NT platforms. They are for tour booking and function room reservation. CICS for AIX region CICKAR8 can access these databases in the same way as the remote DB2 for AIX databases, that is, through the X/Open XA interface and CAE.

To provide comprehensive flight reservation services for hotel guests and other potential travelers, the CICKAR8 region needs to access the Airlines Reservation databases of Kangaroo Airlines that reside on the mainframe computers. This connection is achieved through two separate SNA connections to two MVS/ESA systems from the local RISC System/6000 system running CICKAR8. One connection is through an Encina PPC Gateway, and the other uses CAE and DDCS. The path through CAE and DDCS is the primary route for accessing the Airlines Reservation databases owned by the DB2 for MVS/ESA server DBHQ03. The authentication type of authentication = DCS has been defined for these databases; hence the user name and password are to be validated at the mainframe host. The path through the Encina PPC Gateway is an alternative route to get the same information. Because of the limitation of a maximum size of 32 KB for data passed in the COMMAREA, the use of this route is restricted to queries on Kangaroo Airlines flight schedules.

11.4 File Systems

The majority of TDJ′s enterprise data is held in DB2 databases. The potential for Encina SFS use is therefore limited. TDJ does use Encina SFS to manage accounting data shared by several CICS regions because an Encina SFS server is ideally suited to this sort of application. The Encina SFS server is in the same cell as the CICS regions that access it. Encina SFS servers are used widely by CICS regions such as CICKAR8 to manage CICS queues, although this decision is currently under review.

11.5 Increased Availability

Lion Insurance Limited has become a leading presence in the direct selling of insurance to the public. Customers call a toll-free number and can receive an insurance quotation for car, travel, or house. The customer phone calls are taken by a team of a dedicated sales personnel. During the course of the call, the customer′s details are taken, a quotation is prepared, and the premium calculated. The customer can then decide to accept the quotation or not. Being able to provide such a system calls for a highly available, quick-response quotation system.

To improve the quality of service to the customer, staff other than the dedicated insurance sales team can access the insurance quotation system. Thus it is possible for a customer to receive an insurance quotation without having to be transferred to an insurance agent. Such quotations generate only a small volume of work.

Lion Insurance Limited realizes that selling insurance is very competitive and that high availability of the quotation system is essential to its public image. Lion Insurance has developed a CICS service specifically to meet the need for a high availability, highly performing system. A 16-node RISC System/6000 SP processor provides the processing capacity required for the service. CICS for AIX, DB2 Parallel Edition (PE), and CICS SM provide the key software components. Of the 16 nodes available, CICS for AIX runs on 7 nodes, DB2 PE runs on 7, and 1 node has been allocated as a CICS SM management node. The backup CDS server has been allocated on the remaining node. The backup CDS server will be moved to another machine when the volume of work requires the processing power of an additional node. The application programs which comprise the insurance quotation system are available in each of the CICS for AIX regions. Each of the CICS for AIX regions connects to the parallel database so that data in the database is accessible in each of the nodes. Therefore, if a node is taken out of service for maintenance, the data still remains available. With a partitioned database this would not have been possible. See Figure 45 on page 165 for a view of this environment.

The dedicated sales team uses a CICS for AIX ECI application to interact with the insurance quotation system. This application guides the staff through the necessary questions that they need to ask the customers and then makes an ECI request to a CICS for AIX region to get the quotation processed. As these staff are using CICS for AIX ECI clients, it is possible to workload balance the work generated by using CICS SM workload management. Unfortunately the work from the PC-based users running on CICS on Open Systems clients cannot be workload managed. This volume of work contributes to only a small proportion of the total work. These users are allocated to a separate region, one that is not workload managed, so that the effect of these users does not cause an unexplainable load on the workload-managed systems.

The use of CICS SM workload management ensures that the incoming work is distributed over the available regions. It also copes with the failure of a CICS for AIX region. It does not become a case of some sales staff being unable to work because the region to which they were attached is no longer available, because the incoming work will be distributed over the remaining regions. The use of workload management also means that maintenance can be easily scheduled across the nodes of the SP.

The role of the operations staff has been simplified with the use of CICS SM to manage the insurance quotation systems. Lion Insurance Limited is so pleased that it is looking to implement CICS SM across its other CICS for AIX systems, even those with no requirement to be workload managed. It has also recommended its adoption by the other companies within TDJ.

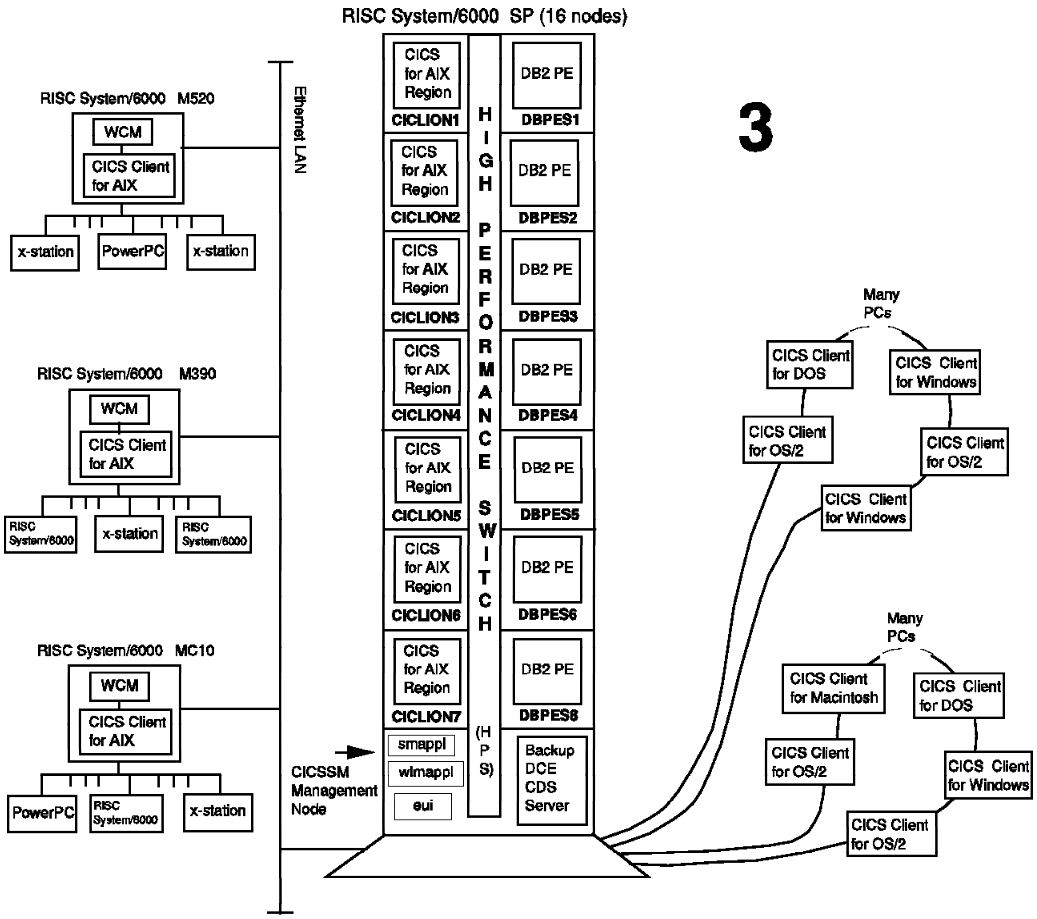

Figure 45. Using CICS System Manager for AIX

11.6 Distributed Processing

Lion Credit Union provides loans of up to $30,000 for a variety of purposes, such as holidays. Requests for loans may come directly to its own dedicated office staff or from a number of authorized agents. Each staff member who deals with loan applications has a PC. The hardware is mixed, coming from multiple suppliers. The operating systems also vary. Each PC runs the appropriate CICS client and is connected to a CICS for OS/2 region. This is a historic connection as the group used to process only staff applications, and the processing was based on a much smaller system. When a decision was made to extend the loan facilities to all credit union members, it was realized that the volume of work was going to be much larger than before. A decision was made to upgrade the new loan system and at the same time move to the AIX platform as part of TDJ′s overall strategy. CICS for AIX region CICSUP9 was created for this purpose. Users now access the loan processing system by transaction routing from their local CICS for OS/2 server.

The authorized agents of Lion Credit Union can also process loan applications. Among them the agents run a variety of different processing hardware. Establishing connectivity to the loan processing region, CICSUP9, is not a problem with CICS. Lion Credit Union runs CICS for AIX, and the agents run the CICS on Open Systems server appropriate to their hardware platform. Communication between Lion Credit Union′s system and the agent′s system takes place over TCP/IP or SNA communications links. Figure 46 on page 166 illustrates the systems and connections. Most of the processing associated with a loan application is performed on the agent's own processor. The people applying for the loan are invariably clients of the agent, and so the data relating to the client is held on the agent's system. Once the agent's system has performed basic checking, a CICS DPL is made to an application in CICSUP9 to get final approval for the loan.

Lion Credit Union staff and the agents are also authorized to provide insurance quotations on behalf of Lion Insurance Limited. Quotation requests are transaction routed to the RISC System/6000 SP system on which the insurance application runs.

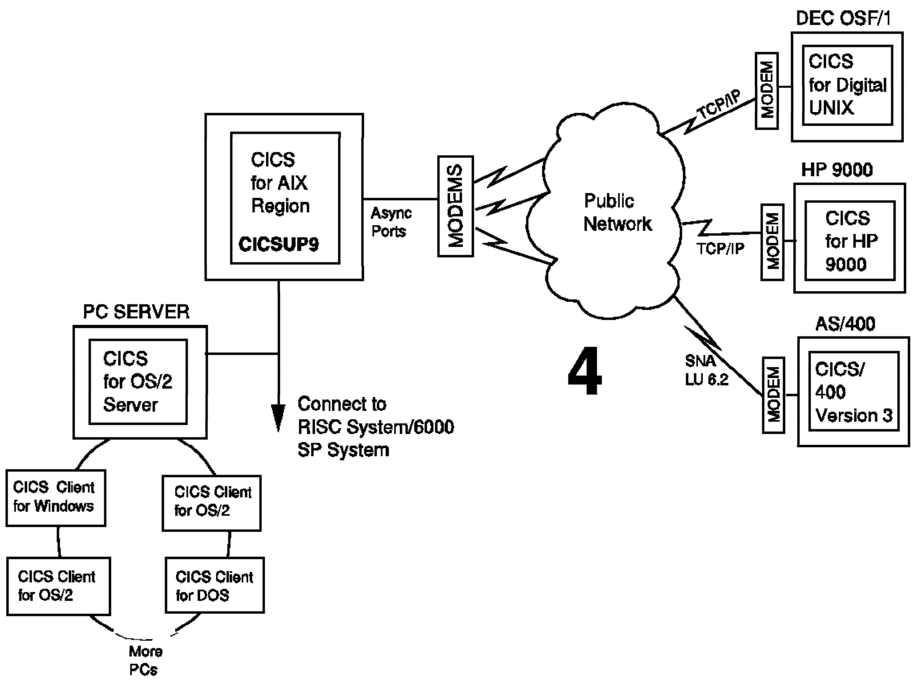

Figure 46. Communicating with Other Members of CICS Family Products

11.7 Internet Access

TDJ, like many corporations, is looking for opportunities to expand its customer base and has recognized the commercial potential of the Internet. Accordingly, TDJ has decided to update its existing applications such as the Accommodation Booking and Guest Invoice Systems with the accessibility and GUI of the Internet. TDJ has just installed the CICS Client for AIX and CICS Internet Gateway in its Web server (RISC System/6000 Model C10 running Internet Connection Server for AIX) so that customers worldwide can access the TDJ Accommodation Booking System using their own Web browsers across the Internet.

Figure 47 on page 167 shows the configuration that services incoming work from the Internet.

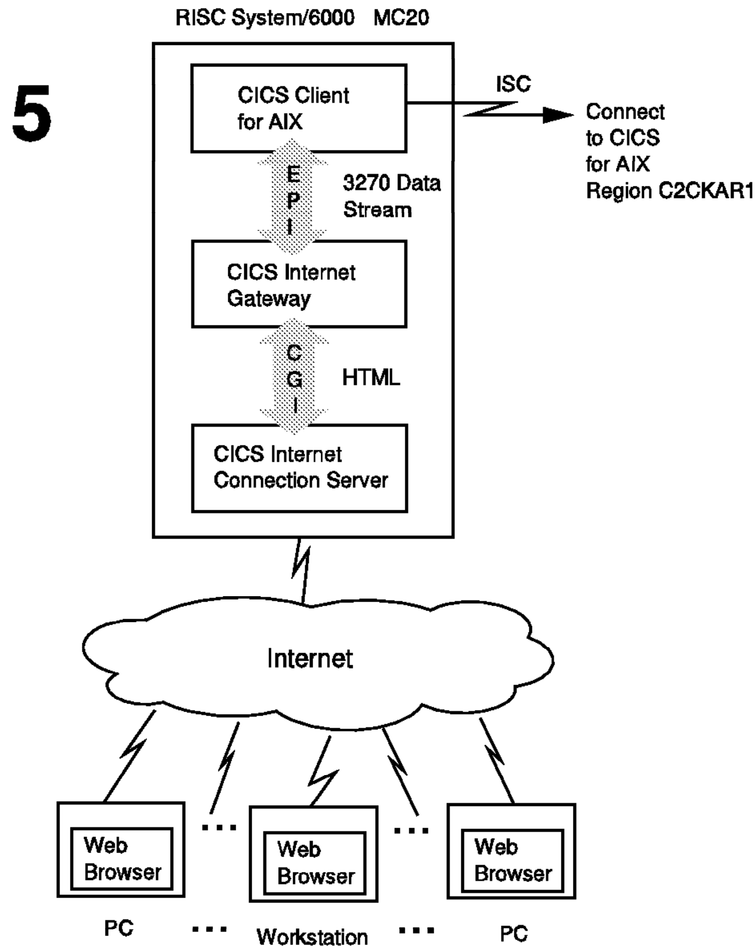

Figure 47. Accessing CICS Applications from the Internet

TDJ has also given some internal CICS applications with no Internet connection, such as the Accounting Control System, a much improved user interface by utilizing the Web browser as an alternative to the 3270 interface. The Web browsers can access any CICS region through the CICS Client running in same machine as the Web server.

11.8 Region Location and Security

There are many CICS regions and thousands of clients in TDJ's network. Security and resource management are important issues for TDJ. Accordingly TDJ has chosen to implement DCE CDS and security. DCE CDS dynamically manages the resolution of CICS regions and SFS server locations. DCE security services are used to authenticate CICS users. CICS internal security is then used additionally to control access to particular CICS resources such as transactions or files.

The DCE servers are located on one node of the 8-node RISC System/6000 SP machine that TDJ owns (see Figure 48 on page 168). Availability is an important issue for TDJ. It does not want systems to be unavailable or inaccessible. Therefore it replicated its DCE CDS server. The replicated DCE CDS server is located on the 16-node RISC System/6000 SP. A separate machine was chosen, rather than another node on the 8-node RISC System/6000 SP machine, to provide greater protection in the unlikely event that the entire 8-node RISC System/6000 SP system experiences an outage.

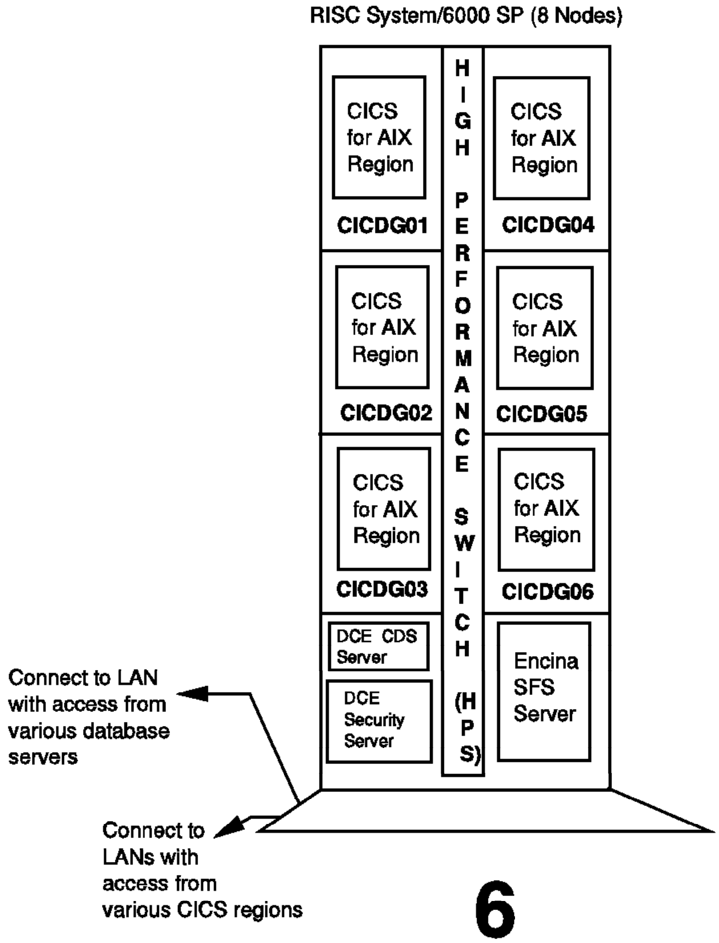

Figure 48. DCE Servers Required by CICS for AIX

11.9 Summary

CICS for AIX in conjunction with the IBM Software Servers have proven themselves capable of handling the most demanding business requirements of TDJ. They can deliver new technologies, such as client-server computing, distributed processing, object-orientation, and the Internet, all of which are constantly evolving in response to changing business needs.

CICS for AIX not only capitalizes on evolving technologies but also enables TDJ to implement client-server solutions across multiple hardware platforms and operating systems. Just as importantly, CICS for AIX frees programmers and end users from the complexities of the underlying technologies and enables them to concentrate on developing and running the necessary business systems. With CICS for AIX, TDJ can make information and applications immediately available where they are most needed—on the desktop of the end user. The end user is not constrained to using only one particular make of PC or running a particular operating system in order to access a CICS for AIX region. The *any CICS client to any CICS server* strategy gives the staff of TDJ the opportunity to exploit business-critical information directly from standard desktop packages, significantly increasing the value of both centralized and local information.

CICS for AIX also protects TDJ′s past investments by extending their usefulness. Through a GUI, the presentation of existing applications has been modernized without having to rewrite the business application. Moreover, mission-critical CICS applications are easily ported to distributed platforms, leveraging the value of the enterprise network.

In the security area, CICS for AIX complements industry-standard security managers with its own security features that coordinate different databases, protect against network failures, and ensure the integrity of business data and application software. TDJ divided its CICS installation on a business-unit basis, providing independent operation for one department while allowing controlled access to resources belonging to another.

CICS for AIX has become the transaction server of choice for TDJ in the new world of distributed, open OLTP. With CICS for AIX, TDJ can deliver on its strategy of a distributed, open OLTP environment across its entire enterprise network.

Appendix A. Special Notices

This publication is intended to help personnel working with IBM Transaction Server for AIX and using CICS for AIX as the transaction monitor. The information in this publication is not intended as the specification of any programming interfaces that are provided by Transaction Server for AIX, CICS for AIX, CICS System Manager for AIX, Encina for AIX, and DCE for AIX. See the PUBLICATIONS section of the IBM Programming Announcement for Transaction Server for AIX, CICS for AIX, CICS System Manager for AIX, Encina for AIX, and DCE for AIX. for more information about what publications are considered to be product documentation.

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

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

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

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

Any performance data contained in this document was determined in a controlled environment, and therefore, the results that may be obtained in other operating environments may vary significantly. Users of this document should verify the applicable data for their specific environment.

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

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

AIX	AIXwindows
C Set ++	CICS
CICS for Windows/NT	CICS for OS/2
CICS/ESA	CICS/MVS
CICS/400	CICS for AIX
DATABASE 2	DB2
DB2 for OS/2	DB2 for AIX
DB2 for MVS/ESA	Enterprise Systems Architecture/390
ES/9000	ESA/390
IBM	MQSeries
MVS/ESA	NetView
OS/2	Personal System/2
PowerPC	PS/2
RACF	RISC System/6000
RS/6000	S/390
SOMobjects	SP2
9076 SP2	

The following terms are trademarks of other companies:

Windows is a trademark of Microsoft Corporation.

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

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

Apple Computer Inc.	Apple, AppleTalk, LocalTalk, Macintosh MacTCP, MPW, System 7, SNA•ps
Hewlett Packard Company	HP
Micro Focus Limited	Micro Focus
Microsoft Corporation	Microsoft, Windows, Windows NT
Motorola	Motorola
Novell, Inc.	Novell, NetWare, LAN Workplace
Open Software Foundation	DCE
Transarc	Encina
Digital Equipment Corporation	DEC, DEC VT100 and DEC VT220
Sun Microsystems	SunOS, SPARCstation, Network File System, NFS, Solaris
Lotus Development Corporation	Lotus Notes

Other trademarks are trademarks of their respective companies.

Appendix B. Related Publications

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

B.1 International Technical Support Organization Publications

For information on ordering these ITSO publications see "How To Get ITSO Redbooks" on page 175.

- *CICS Clients Unmasked*, SG24-2534-01
- *Distributed Relational Database Cross Platform Connectivity and Application*, GG24-4311
- *AIX CICS/6000 and Relational Database Management Systems Integration: Experiences with the XA Interface*, GG24-4214
- *Accessing CICS Business Applications from the World Wide Web* , SG24-4547-01

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

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

B.2 Redbooks on CD-ROMs

Redbooks are also available on CD-ROMs. **Order a subscription** and receive updates 2-4 times a year at significant savings.

CD-ROM Title	Subscription Number	Collection Kit Number
System/390 Redbooks Collection	SBOF-7201	SK2T-2177
Networking and Systems Management Redbooks Collection	SBOF-7370	SK2T-6022
Transaction Processing and Data Management Redbook	SBOF-7240	SK2T-8038
AS/400 Redbooks Collection	SBOF-7270	SK2T-2849
RISC System/6000 Redbooks Collection (HTML, BkMgr)	SBOF-7230	SK2T-8040
RISC System/6000 Redbooks Collection (PostScript)	SBOF-7205	SK2T-8041
Application Development Redbooks Collection (available soon)	SBOF-7290	SK2T-8037
Personal Systems Redbooks Collection (available soon)	SBOF-7250	SK2T-8042

B.3 Other Publications

These publications are also relevant as further information sources:

- J.Gray and A.Reuter, *Transaction Processing: Concepts and Techniques*. San Mateo (CA): Morgan Kaufman Publishers (1993)
- "IBM Transaction Server Marketplace and Competition," IBM White Paper, IBM Corporation (1996)
- *CICS Family: General Information Manual*, GC33-0155-05
- *CICS Family: API Structure*, SC33-1007-01
- *CICS on Open Systems V2.1 Administration Guide*, SC33-1563-00

- *CICS on Open Systems V2.1.1 Application Programming Reference*, SC33-1569-01
- *CICS on Open Systems Application Programming Guide*, SC33-1568-01
- *CICS for AIX Planning and Installation Guide*, GC33-1773-00
- *CICS/ESA Dynamic Transaction Routing in a CICSplex*, SC33-1012-00
- *CICS Family: Client/Server Programming*, SC33-1435-00
- *CICS Clients Administration, Version 1.1*, SC33-1436-01
- *CICS Interproduct Communication*, SC33-0824-02
- *CICS on Open Systems Intercommunication Guide*, SC33-1564-01
- *IBM CICS System Manager for AIX V1.1 Product Overview*, GC33-1592-00
- *IBM CICS System Manager for AIX V1.1 User's Guide*, SC33-1594-00
- *DB2 for AIX Planning Guide*, S20H-4758-00
- *DB2 for AIX Installation and Operation*, S20H-4757-00
- *DB2 for AIX Administration Guide*, S20H-4580-00
- *DDCS User's Guide*, S20H-4793-00
- *DRDA Planning for Distributed Relational Databases*, SC26-4650-01
- *SNA Server for AIX: Planning and Performance Guide*, SC31-8220-00
- *X/Open Distributed Transaction Processing Reference Model*, document number G120, ISBN 1 872630 16 2

How To Get ITSO Redbooks

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

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

How IBM Employees Can Get ITSO Redbooks

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

- **PUBORDER** — to order hardcopies in United States
- **GOPHER link to the Internet** - type GOPHER.WTSCPOK.ITSO.IBM.COM
- **Tools disks**

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

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

  To get lists of redbooks:

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

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

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

  For a list of product area specialists in the ITSO:

  ```
  TOOLS SENDTO WTSCPOK TOOLS ZDISK GET ORGCARD PACKAGE
  ```

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

  http://w3.itso.ibm.com/redbooks

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

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

  IBM employees may obtain LIST3820s of redbooks from this page.

- **ITSO4USA category on INEWS**
- **Online** — send orders to: USIB6FPL at IBMMAIL or DKIBMBSH at IBMMAIL
- **Internet Listserver**

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

How Customers Can Get ITSO Redbooks

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

- **Online Orders** (Do not send credit card information over the Internet) — send orders to:

	IBMMAIL	**Internet**
In United States:	usib6fpl at ibmmail	usib6fpl@ibmmail.com
In Canada:	caibmbkz at ibmmail	lmannix@vnet.ibm.com
Outside North America:	bookshop at dkibmbsh at ibmmail	bookshop@dk.ibm.com

- **Telephone orders**

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

- **Mail Orders** — send orders to:

IBM Publications	IBM Publications	IBM Direct Services
Publications Customer Support	144-4th Avenue, S.W.	Sortemosevej 21
P.O. Box 29570	Calgary, Alberta T2P 3N5	DK-3450 Allerød
Raleigh, NC 27626-0570	Canada	Denmark
USA		

- **Fax** — send orders to:

United States (toll free)	1-800-445-9269
Canada	1-403-267-4455
Outside North America	(+45) 48 14 2207 (long distance charge)

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

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

- **Direct Services** - send note to softwareshop@vnet.ibm.com

- **On the World Wide Web**

  Redbooks Home Page http://www.redbooks.ibm.com
  IBM Direct Publications Catalog http://www.elink.ibmlink.ibm.com/pbl/pbl

- **Internet Listserver**

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

IBM Redbook Order Form

Please send me the following:

Title	Order Number	Quantity

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

First name Last name

Company

Address

City Postal code Country

Telephone number Telefax number VAT number

- Invoice to customer number

- Credit card number

Credit card expiration date Card issued to Signature

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

DO NOT SEND CREDIT CARD INFORMATION OVER THE INTERNET.

Glossary

A

abend. Abnormal ending of transaction.

API. Application programming interface. A set of calling conventions defining how a service is invoked through a software package.

APPC. Advanced Program-to-Program Communication. An implementation of SNA′s LU 6.2 protocol that allows interconnected systems to communicate and share the processing of programs.

application manager. A CICS run-time process that manages a pool of application servers.

application server. A CICS run-time process that executes CICS tasks.

application unit of work. A set of actions within an application that the designer chooses to regard as an entity. It is up to the designer to decide how, if at all, an application should be divided into application units of work, and whether any application unit of work will consist of one, or many, logical units of work (LUWs). Typically, but not exclusively, an application unit of work corresponds to a CICS transaction.

asynchronous. Without regular time relationship; unexpected or unpredictable with respect to the execution of program instruction. See *synchronous*.

B

business process. An entity-handling activity that is of limited duration, defined in scope, and set by business goals and policies, not by organization or implementation.

C

CEMT. A CICS-supplied transaction that invokes all master terminal functions. These functions include inquiring about and changing the value of parameters used by CICS, altering the status of system resources, and terminating tasks.

cicsprnt. A CICS-supplied program that provides the ability to define a printer terminal on the client workstation. The program enables CICS applications running on the server to direct output to the client-attached printer. It is supplied with the CICS Clients.

cicsterm. A CICS-supplied program that provides 3270 emulation and enables connection to a CICS region. It is provided as part of the CICS Clients.

CICS. Customer Information Control System (CICS). A distributed online transaction processing system—an online system controller and some utilities that are capable of supporting a network of many terminals. The CICS family of products provides a range of application platforms on many operating system platforms.

client. As in client-server computing, the application that makes requests to the server and, often, deals with the interaction necessary with the user.

client-server computing. A form of distributed processing, in which the task required to be processed is accomplished by a client portion that requests services and a server portion that fulfills those requests. The client and server remain transparent to each other in terms of location and platform. See *client, distributed processing, and server*.

commit. An action that an application takes to make permanent the changes it has made to CICS resources.

conversational. Communication model where two distributed applications exchange information by way of a conversation; typically one application starts (or allocates) the conversation, sends some data, and allows the other application to send some data. Both applications continue in turn until one decides to finish (or deallocate). The conversational model is a synchronous form of communication.

cooperative processing. The process by which a single application is divided between two or more hardware platforms. Very often the term is used to reflect a tightly coupled relationship between the parts of the application.

D

database. (1) A collection of interrelated data stored together with controlled redundancy according to a scheme to serve one or more applications. (2) All data files stored in the system. (3) A set of data stored together and managed by a database management system.

DCE. Distributed Computing Environment. Adopted by the computer industry as a de facto standard for distributed computing. DCE allows computers from a variety of vendors to communicate transparently and share resources such as computing power, files, printers, and other objects in the network.

design. The process of composing a software blueprint, showing how to build from the requirements specification document. Design often includes module decompositions, data structure definitions, file format definitions, and important algorithm descriptions.

distributed processing. Distributed processing is an application or systems model in which function and data can be distributed across multiple computing resources connected on a LAN or WAN. See *client-server computing*.

DPL. Distributed program link. Provides a remote procedure call (RPC)-like mechanism by function shipping EXEC CICS LINK commands. In CICS/6000, DPL allows the linked-to program to issue unsupported CICS/6000 function shipping calls, such as to DB2 and DL/I, and yields performance improvements for transactions that read many records from remote files.

DTP. Distributed transaction processing. Type of intercommunication in CICS. The processing is distributed between transactions that communicate synchronously with one another over intersystem links. DTP enables a CICS application program to initiate transaction processing in a system that supports LU 6.2 and resides in the same or a different processor.

E

ECI. external call interface. An application programming interface (API) that enables a non-CICS client application to call a CICS program as a subroutine. The client application communicates with the server CICS program, using a data area called a COMMAREA.

Encina. Enterprise computing in a new age. A set of DCE-based products from Transarc Corporation that are available on the RISC System/6000. The Encina family of online transaction processing products includes:

- Encina Toolkit Executive
- Encina Server
- Encina Structured File Server (SFS)
- Encina Peer-to-Peer Communication Executive (PPC).

EPI. External presentation interface. An application programming interface (API) that allows a non-CICS application program to appear to the CICS system as one or more standard 3270 terminals. The non-CICS application can start CICS transactions and send and receive standard 3270 data streams to those transactions.

environment. The collective hardware and software configuration of a system.

F

file server. A centrally located computer that acts as a storehouse of data and applications for numerous users of a local area network.

G

GUI. Graphical user interface. A style of user interface that replaces the character-based screen with an all-points-addressable, high-resolution graphics screen. Windows display multiple applications at the same time and allow user input by means of a keyboard or a pointing device such as a mouse, pen, or trackball.

H

heterogeneous network. A local or wide area network comprising hardware and software from different vendors, usually implementing multiple protocols, operating systems, and applications.

host. (1) In a computer network, a computer providing services such as computation, database access, and network control functions. (2) The primary or controlling computer in a multiple computer installation.

I

instance. An individual application object that belongs to a particular object class. An instance of a class is an actual object displaying the properties of that class.

intercommunication. Communication between separate systems by means of Systems Network Architecture (SNA), Transmission Control Protocol/Internet Protocol (TCP/IP), and Network Basic Input/Output System (NetBIOS) networking facilities.

interoperability. The ability to interconnect systems from different manufacturers and have them work together to satisfy a business requirement. Some examples of requirements are message interchange between systems and sharing of resources such as data between applications running on different hardware and software platforms.

L

LU type 6.2 (LU 6.2). Type of logical unit used for CICS intersystem communication (ISC). The LU 6.2 architecture supports CICS host to system-level products and CICS host to device-level products. APPC is the protocol boundary of the LU 6.2 architecture.

LUW. Logical unit of work. An update that durably transforms a resource from one consistent state to another consistent state. A sequence of processing actions (for example, database changes) that must be completed before any of the individual actions can be regarded as committed. When changes are committed (by successful completion of the LUW and recording of the syncpoint on the system log), they do not have to be backed out after a subsequent error within the task or region. The end of an LUW is marked in a transaction by a syncpoint that is issued by either the user program or the CICS server, at the end of task. If there are no user syncpoints, the entire task is an LUW.

M

macro. An instruction that causes the execution of a predefined sequence of instructions in the source language in which the macro is embedded. The predefined sequence can be modified by parameters in the macro.

messaging. A communications model whereby the distributed applications communicate by sending messages to each other. A message is typically a short packet of information that does not necessarily require a reply. Messaging implements asynchronous communications.

middleware. Middleware is a set of services that allows distributed applications to interoperate on a LAN or WAN. It shields the developer or end user from the system complexity and enables delivery of service requests or responses transparently across computing resources.

O

object. A program or a group of data that can behave like a thing in the real world.

OLTP. Online transaction processing. A style of computing that supports interactive applications in which requests submitted by terminal users are processed as soon as they are received. Results are returned to the requester in a relatively short period of time. An online transaction processing system supervises the sharing of resources for processing multiple transactions at the same time, minimizes compute time and duration of locks, and separates user think-time from the use of storage and other resources.

P

partner LU. In SNA, one of an LU pair between which a session is established.

portability. The ability to move application software components from one system for use on another system. Perfect portability would permit such movement without modification of the components.

process. (1) A unique, finite course of events defined by its purpose or by its effect, achieved under defined conditions. (2) Any operation or combination of operations on data. (3) A function being performed or waiting to be performed. (4) A program in operation, for example, a daemon.

protocol. (1) A formal set of conventions governing the format and control of data. (2) A set of procedures or rules for establishing and controlling transmissions from a source device or process to a target device or process.

pseudoconversational. A type of CICS application design that appears to the user as a continuous conversation but consists internally of multiple tasks.

R

recovery. The use of archived copies to reconstruct files, databases, or complete disk images after they are lost or destroyed.

recoverable resources. Items whose integrity CICS maintains in the event of a system error. These include individual files and queues.

RPC. Remote procedure call. A communication model where requests are made by function calls to distributed procedures elsewhere. The location of the procedures is transparent to the calling application.

resource manager. A software program that maintains the state of resources and provides access and control to them through APIs. A resource can be a device as well as a program or object, although normally it is referred to as a device.

S

server. Any computing resource dedicated to responding to client requests. Servers can be linked to clients through LANs or WANs to perform services, such as printing, database access, fax, and image processing, on behalf of multiple clients at the same time.

SQL. Structured query language. SQL started as IBM′s query language for DB2. SQL became so popular with users and vendors outside IBM that ANSI adopted a version of SQL as a U.S. standard in 1986.

A year later ISC gave SQL formal international standard status.

stored procedures. Facility for storing procedural code associated with relational database management systems (RDBMSs) that enforces its use during any database operation.

synchronous. (1) Pertaining to two or more processes that depend on the occurrence of a specific event such as a common timing signal. (2) Occurring with a regular or predictable time relationship.

sync point. A logical point in execution of an application program where the changes made to the databases by the program are consistent and complete and can be committed to the database. The output, which has been held up to that point, is sent to its destination, the input is removed from the message queues, and the database updates are made available to other applications. When a program terminates abnormally, CICS recovery and restart facilities do not back out updates made before the last completed sync point.

T

test. Testing involves checking each individual module built during the implementation phase, then integrating the modules into a single program structure. The program as a whole is then tested to ensure that it performs as designed.

thread of control. Inside the X/Open DTP model, the concept that associates resource manager work with the global transaction. Routines in the XA interface that manage the association between a thread of control and transactions must be called from the same thread.

transaction. A unit of processing (consisting of one or more application programs) initiated by a single request. A transaction can require the initiation of one or more tasks for its execution.

transaction manager. Provides the function to begin, end, commit, and roll back transactions.

transaction branch. A part of the work in support of a global transaction for which the transaction manager and the resource manager engage in a commitment protocol coordinated with, but separate from, that for other branches.

transaction manager. Provides the function to begin, end, commit, and roll back transactions.

transaction monitor. Provides a total environment for transactional applications. In addition to transaction manager functions, provides services to aid development, execution, and operation of transaction applications.

transaction processing. A style of computing that supports interactive applications in which requests submitted by users are processed as soon as they are received. Results are returned to the requester in a relatively short period of time. A transaction processing system supervises the sharing of resources for processing multiple transactions at the same time.

two-phase commit. For a database, a protocol that is used to ensure uniform transaction commit or abort in a distributed data environment between two or more participants. The protocol consists of two phases: the first to reach a common decision, and the second to implement the decision.

TX. Within the DTP model adopted by X/Open, the interface among the application or transaction monitor and the transaction manager.

W

workstation. A configuration of input/output equipment at which an operator works. A terminal or microcomputer, usually one that is connected to a mainframe or a network, at which a user can perform applications.

X

XA. Within the DTP model adopted by X/Open, the interface between the transaction manager and resource managers.

List of Abbreviations

ACL	access control list
AID	attention identifier
AIX	Advance Interactive Executive
APPC	Advanced Program-to-Program Communication
API	application programming interface
BMS	basic mapping support
CAE	client application enabler
CDS	cell directory service
CEDF	CICS Execution Diagnostic Facility
CGI	common gateway interface
CICS	Customer Control Information System
CICS SM	CICS System Manager for AIX
CWA	common work area
DB2	DATABASE 2
DCE	Distributed Computing Environment
DDCS	Distributed Database Connection Services
DRDA	Distributed Relational Database Architecture
DSOM	Distributed System Object Model
DTS	distributed time service
DUW	distributed unit of work
ECI	external call interface
EIB	enterprise information base
Encina	Enterprise Computing in New Age
EPI	external presentation interface
GDS	global directory service
GUI	graphical user interface
HTML	Hyper Text Makeup Language
HTTP	Hyper Text Transfer Protocol
IBM	International Business Machines Corporation
ITSO	International Technical Support Organization
IT	information technology
LD	listener definition
LUW	logical unit of work
MCA	message channel agent
MQI	message queuing interface
OFD	SFS open file descriptor
OLTP	online transaction processing
OSF	Open Software Foundation
RDBMS	relational database management system
RDS	remote data service
RPC	remote procedure call
SDA	screen design aid
SFS	Encina Structured File Server
SMAppl	system management application
SPOC	single point of control
SSI	single systems image
TPC	Transaction Processing Council
TPS	transactions per second
URL	uniform resource locator
WLMAppl	workload management application
WWW	World Wide Web
XAD	XA definition

Index

U

W

X

Printed in U.S.A.

SG24-4752-00